目 录

第1章 MPLS 体系结构

1.1 MPLS 概述

MPLS（Multiprotocol Label Switching）是多协议标签交换的简称，它用短而定长的标签来封装网络层分组。MPLS 从各种链路层（如 PPP、ATM、帧中继、以太网等）得到链路层服务，又为网络层提供面向连接的服务。MPLS 能从 IP 路由协议和控制协议中得到支持，同时，还支持基于策略的约束路由，它路由功能强大、灵活，可以满足各种新应用对网络的要求。这种技术起源于 IPv4，但其核心技术可扩展到多种网络协议（IPv6、IPX 等）。

MPLS 最初是为提高路由器的转发速度而提出一个协议，但是，它的用途已不仅仅局限于此，而是广泛地应用于流量工程（Traffic Engineering）、VPN、QoS 等方面，从而日益成为大规模 IP 网络的重要标准。

1.2 基本概念

1.2.1 转发等价类（FEC）

FEC（Forwarding Equivalence Class）是 MPLS 中的一个重要概念。MPLS 实际上是一种分类转发技术，它将具有相同转发处理方式（目的地相同、使用转发路径相同、具有相同的服务等级等）的分组归为一类，称为转发等价类。一般来说，划分分组的 FEC 是根据他的网络层目的地址。属于相同转发等价类的分组在 MPLS 网络中将获得完全相同的处理。

1.2.2 标签

1. 标签的定义

标签为一个长度固定、具有本地意义的短标识符，用于标识一个 FEC（Forwarding Equivalence Class）。当分组到达 MPLS 网络入口时，它将按一定规则被划归不同的 FEC，根据分组所属的 FEC，将相应的标签封装在分组中，这样，在网络中，按标签进行转发分组即可。

2. 标签的结构

标签的结构如 图 1-1所示。

Label	Exp	S	TTL

图1-1 标签的结构

标签位于链路层包头和网络层分组之间，长度为 4 个字节。标签共有 4 个域：

Label：标签值字段，长度为 20bits，用于转发的指针。

Exp：3bits，保留，用于试验。

S：1bit，MPLS 支持标签的分层结构，即多重标签。值为 1 时表明为最底层标签。

TTL：8bits，和 IP 分组中的 TTL 意义相同。

3. 标签的操作

(1) 标签映射

标签映射分为两种，一种是入口路由器处的标签映射，另一种是 MPLS 域内的标签映射。

入口路由器处的标签映射为 ingress LSR 依据一定的原则对输入分组进行划分，得到多个 FEC，接着将有关标签与这些 FEC 进行映射，并记录在相应的数据库 LIB（Label Information Base）中。简单地说，就是将一个标签指派给 FEC，就称为“标签映射”。

MPLS 域内的标签映射又称为输入标签映射 ILM（Incoming Label Map），即将每个输入标签映射到一系列 NHLFE（Next Hop Label Forwarding Entry）上，然后，根据映射结果，将分组沿各通路进行转发。

(2) 标签的封装

标签在各种介质中的封装如下图所示：

以太网/SONET/SDH分组	以太网报头/PPP报头	标签	三层数据
帧模式ATM分组	ATM报头	标签	三层数据
信元模式的ATM分组	VPI/VCI		三层数据

图1-2 标签在分组中的封装位置

对于以太网、PPP 的分组，标签堆栈象“垫层”一样，位于二层报头与数据之间，对于 ATM 信元模式的分组，借用 VPI/VCI 来作为标签使用。

(3) 标签分配和分发

标签分发是为某 FEC 建立相应标签交换路径 LSP 的过程。

在 MPLS 体系中，将特定标签分配给特定 FEC（即标签绑定）的决定由下游 LSR 作出，下游 LSR 随后通知上游 LSR。即标签由下游指定，分配的标签按照从下游到上游的方向分发。

MPLS 中使用的标签分发方式有两种：下游自主标签分发方式（DU，Downstream Unsolicited）和下游按需标签分发方式（DoD，Downstream On Demand）。

- 对于一个特定的 FEC，LSR 无须从上游获得标签请求消息即进行标签分配与分发的方式，称为下游自主标签分配。
- 对于一个特定的 FEC，LSR 获得标签请求消息之后才进行标签分配与分发的方式，称为下游按需标签分配。

具有标签分发邻接关系的上游 LSR 和下游 LSR 之间，必须对使用哪种标签分发方式达成一致。

LSR 将标签分发给其对等体时，可以采用 LDP 消息进行传送，也可以将标签搭载在其他路由协议消息上。

说明：

上游和下游是相对而言的，对于一个报文转发过程来说，发送方的路由器是上游 LSR，接收方是下游 LSR。

---

(4) 标签分配控制方式

标签分配控制方式分为两种：独立（Independent）标签分配控制方式和有序（ordered）标签分配控制方式。

当使用独立标签分配控制方式时，每个 LSR 可以在任意时间向和它连接的 LSR 通告标签映射。

当使用有序标签分配控制方式时，只有当 LSR 收到某一特定 FEC 下一跳的特定标签映射消息或者 LSR 是 LSP 的出口节点时，LSR 才可以向上游发送标签映射消息。

(5) 标签保持方式

标签保持方式分为两种：自由标签保持方式和保守标签保持方式。

假设两台路由器 Ru，Rd，对于特定的一个 FEC，如果 LSR Ru 收到了来自 LSR Rd 的标签绑定，当 Rd 不是 Ru 的下一跳时，如果 Ru 保存该绑定，则称 Ru 使用的是自由标签保持方式；如果 Ru 丢弃该绑定，则称 Ru 使用的是保守标签保持方式。

当要求 LSR 能够迅速适应路由变化时，可使用自由标签保持方式；当要求 LSR 中保存较少的标签数量时，可使用保守标签保持方式。

1.2.3 标签分发协议（LDP）

标签分发协议 LDP（Label Distribution Protocol）是 MPLS 中的信令控制协议，是控制 LSR 之间交换标签与 FEC 绑定，协调 LSR 间工作的一系列规程。

1.3 MPLS 体系结构

1.3.1 MPLS 网络结构

MPLS 网络的基本构成单元是标签交换路由器 LSR（Label Switching Router），主要运行 MPLS 控制协议和第三层路由协议，并负责与其他 LSR 交换路由信息来建立路由表，实现 FEC 和 IP 分组头的映射，建立 FEC 和标签之间的绑定，分发标签绑定信息，建立和维护标签转发表等工作。

由 LSR 构成的网络叫做 MPLS 域，位于区域边缘的 LSR 称为边缘 LSR（LER，Labeled Edge Router），主要完成连接 MPLS 域和非 MPLS 域以及不同 MPLS 域的功能，并实现对业务的分类、分发标签（作为出口 LER）、剥去标签等。其中入口 LER 叫 Ingress，出口 LER 叫 Egress。

位于区域内部的 LSR 则称为核心 LSR，核心 LSR 可以是支持 MPLS 的路由器，也可以是由 ATM 交换机等升级而成的 ATM-LSR，它提供标签交换（Label Swapping）标签分发、交换功能。带标签的分组沿着由一系列 LSR 构成的标签交换路径 LSP（Label Switched Path）传送。

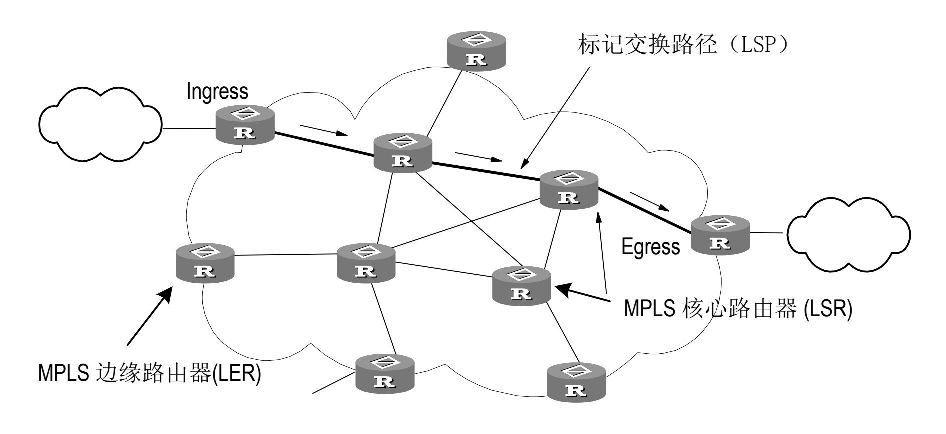


图1-3 MPLS 基本原理

1.3.2 标签报文的转发

在 Ingress，将进入网络的分组根据其特征划分成转发等价类 FEC。一般根据 IP 地址前缀或者主机地址来划分 FEC。属于相同 FEC 的分组在 MPLS 区域中将经过相

同的路径（即 LSP）。LSR 对到来的 FEC 分组分配一个短而定长的标签，然后从相应的接口转发出去。

在 LSP 沿途的 LSR 上，都已建立了输入/输出标签的映射表（该表的元素叫下一跳标签转发条目，简称 NHLFE，Next Hop Label Forwarding Entry）。对于接收到的标签分组，LSR 只需根据标签从表中找到相应的 NHLFE，并用新的标签来替换原来的标签，然后，对标签分组进行转发，这个过程叫输入标签映射 ILM（Incoming Label Map）。

MPLS 在网络入口处指定特定分组的 FEC，后续路由器只需简单的转发即可，比常规的网络层转发要简单得多，转发速度得以提高。

说明：

TTL 处理：
标签化分组时必须将原 IP 分组中的 TTL 值拷贝到标签中的 TTL 域。LSR 在转发标签化分组时，要对栈顶标签的 TTL 域作减一操作。标签出栈时，再将栈顶的 TTL 值拷贝回 IP 分组或下层标签。
但是，当 LSP 穿越由 ATM-LSR 或 FR-LSR 构成的非 TTL LSP 段时，域内的 LSR 无法处理 TTL 域。这时，需要在进入非 TTL LSP 段时对 TTL 进行统一处理，即一次性减去反映该非 TTL LSP 段长度的值。

1.3.3 LSP 的建立

LSP 的建立其实就是将 FEC 和标签进行绑定，并将这种绑定通告 LSP 上相邻 LSR 的过程。这个过程是通过标签分发协议 LDP 来实现的。LDP 规定了 LSR 间的消息交互过程和消息结构，以及路由选择方式。

1. LDP 的工作过程

LSR通过周期性地发送Hello消息来发现LSR邻居，然后与新发现的相邻LSR间建立LDP会话。通过LDP会话，相邻LSR间通告标签交换方式、标签空间、会话保持定时器值等信息。LDP会话是TCP连接，需通过LDP消息来维护，如果在会话保持定时器值规定的时间内没有其它LDP消息，那么必须发送会话保持消息来维持LDP会话的存在。图 1-4为LDP标签分发示意图。

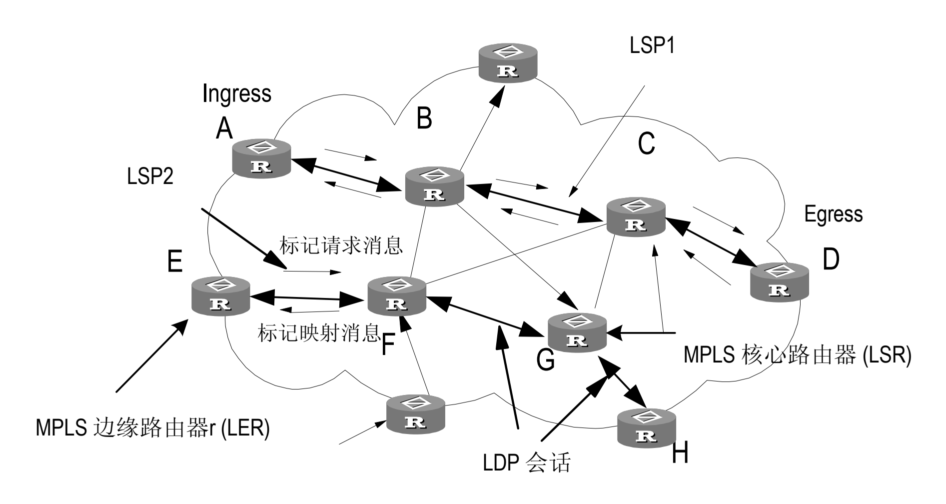


图1-4 标签分发过程

前面提到，标签的分发过程有两种模式：DoD（downstream-on-demand）模式和DU（downstream unsolicited）模式。这两种模式的主要区别在于标签映射的发布是上游请求还是下游主动发布。

DoD模式下标签的分发过程是这样：上游LSR向下游LSR发送标签请求消息（包含FEC的描述信息），下游LSR为此FEC分配标签，并将绑定的标签通过标签映射消息反馈给上游LSR。下游LSR在何时反馈标签映射消息，取决于该LSR采用独立标签分配控制方式还是有序标签分配控制方式。当下游LSR采用有序标签分配控制方式时，只有收到它的下游返回的标签映射消息后，才向其上游发送标签映射消息；当下游LSR采用独立标签分配控制方式时，则不管有没有收到它的下游返回的标签映射消息，都立即向其上游发送标签映射消息。上游LSR一般是根据其路由表中的信息来选择下游LSR的。图 1-4中LSP1 沿途的LSR都采用有序标签分配控制方式，LSP2 上LSR F采用独立标签分配控制方式。

DU（downstream unsolicited）模式下分发标签的过程：下游 LSR 在 LDP 会话建立成功，主动向其上游 LSR 发布标签映射消息。上游 LSR 保存标签映射信息，并根据路由表信息来处理收到的标签映射信息。

2. 基于约束路由的 LDP

MPLS 还支持基于约束路由的 LDP 机制（CR-LDP，Constrain-based Routing LDP）。所谓 CR-LDP，就是入口节点在发起建立 LSP 时，在标签请求消息中对 LSP 路由附加了一定的约束信息。这些约束信息可以是对沿途 LSR 的精确指定，此时叫严格的显式路由；也可以是对选择下游 LSR 的模糊限制，此时叫松散的显式路由。

3. LSP 的环路控制

在 MPLS 域中建立 LSP 也要防止路径循环。防止 LSP 的路径循环有最大跳数和路径向量两种方式。

最大跳数方式是在传递标签绑定的消息中包含跳数信息，每经过一跳该值就加 1，当该值超过规定的最大值时，就认为出现了环路，从而终止 LSP 的建立过程。

路径向量方式是在传递标签绑定的消息中记录路径信息，每经过一跳，相应的路由器就检查自己的 ID 是否在此记录中，如果没有，就将自己的 ID 添加到该记录中；若有，就说明出现了环路，终止 LSP 的建立过程。

1.3.4 LSP 隧道与分层

1. LSP 隧道

MPLS 支持 LSP 隧道技术。在一条 LSP 路径上，LSR Ru 和 LSR Rd 互为上下游，但LSR Ru和LSR Rd之间的路径，可能并不是路由协议所提供路径的一部分，MPLS允许在 LSR Ru 和 LSR Rd 间建立一条新的 LSP 路径<Ru R1...Rn Rd>，LSR Ru 和 LSR Rd 分别为这条 LSP 的起点和终点。LSR Ru 和 LSR Rd 间的 LSP 就是 LSP 隧道，它避免了传统的网络层封装隧道。当隧道经由的路由和逐跳与从路由协议取得的路由一致时，这种隧道叫逐跳路由隧道；若不一致，则这种隧道叫显式路由隧道。

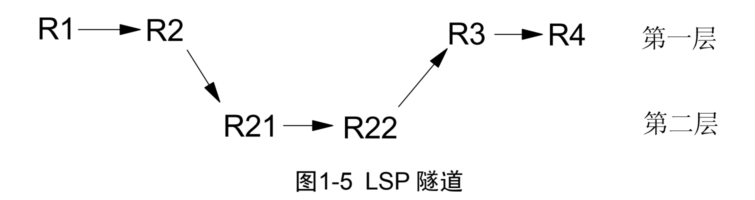


图1-5 LSP 隧道

在上图中，LSP<R2 R21 R22 R3>就是 R2、R3 间的一条隧道。

2. 多层标签栈

在 MPLS 中，分组可以携带多个标签，这些标签在分组中以“堆栈”的形式存在，对堆栈的操作按 “后进先出”的原则，决定如何转发分组的标签始终是栈顶标签。标签入栈是指向输出分组中加入一个标签，使标签栈的深度加 1，同时，分组的当前标签就变为此新加入的标签；标签出栈是指从分组中去掉一个标签，使标签栈的深度减 1，同时，分组的当前标签将变为原来处于下一层的标签。

在 LSP 隧道中会使用多层标签栈。当分组在 LSP 隧道中传送时，分组的标签就会有多层。在每一隧道的入口和出口处，要进行标签栈的入栈和出栈操作，每发生一次入栈操作，标签就会增加一层。MPLS 对标签栈的深度没有限制。

标签栈按照“后进先出”方式组织标签，MPLS 从栈顶开始处理标签。

若一个分组的标签栈深度为 m，则位于栈底的标签为 1 级标签，位于栈顶的标签为 m 级标签。未打标签的分组可看作标签栈为空（即标签栈深度为零）的分组。

1.4 MPLS 与其他协议间的关系

1.4.1 MPLS 与路由协议的关系

LDP 通过逐跳方式建立 LSP 时，要利用沿途各 LSR 路由转发表中的信息来确定下一跳，而路由转发表中的信息一般是通过 IGP、BGP 等路由协议收集的。但是，LDP 并不直接和各种路由协议有关联，只是间接使用路由信息。

另一方面，虽然 LDP 是专门用来实现标签分发的协议，但 LDP 并不是唯一的标签分发协议。对 BGP、RSVP 等已有协议进行扩展，也可以支持 MPLS 标签的分发。MPLS 的一些应用也需要对某些路由协议进行扩展。例如，基于 MPLS 的 VPN 应用就需要对 BGP 协议进行扩展，以便 BGP 协议能传播 VPN 的路由信息；基于 MPLS 的流量工程 TE（Traffic Engineering）需要对 OSPF 或 IS-IS 协议进行扩展，以便携带链路状态信息。

1.4.2 RSVP 对 MPLS 的扩展

资源预留协议 RSVP（Resource Reservation Protocol）经扩展后可以支持 MPLS 标签的分发，同时，在传送标签绑定消息时，还能携带资源预留的信息。通过这种方法建立的 LSP 可以具有资源预留功能，即沿途的 LSR 可以为该 LSP 分配一定的资源，使在此 LSP 上传送的业务得到保证。

RSVP 协议的扩展主要是在其 Path 消息和 Resv 消息中增加新的对象，这些新对象除了可以携带标签绑定信息外，还可以携带对沿途 LSR 寻径时的限制信息，从而支持 LSP 约束路由的功能。扩展的 RSVP 协议还支持快速重路由，即在一定条件下 LSP 需要改变时，可以在不中断用户业务的同时，将原来的业务流重新路由到新建立的 LSP 上。

1.5 MPLS 应用

1.5.1 基于 MPLS 的 VPN

传统的 VPN 一般是通过 GRE、L2TP、PPTP 等隧道协议来实现私有网络间数据流在公网上的传送。LSP 本身就是公网上的隧道，用 MPLS 来实现 VPN 有天然的优势。基于 MPLS 的 VPN 就是通过 LSP 将私有网络在地域上的不同分支联结起来，形成一个统一的网络。基于 MPLS 的 VPN 还支持不同 VPN 间的互通。

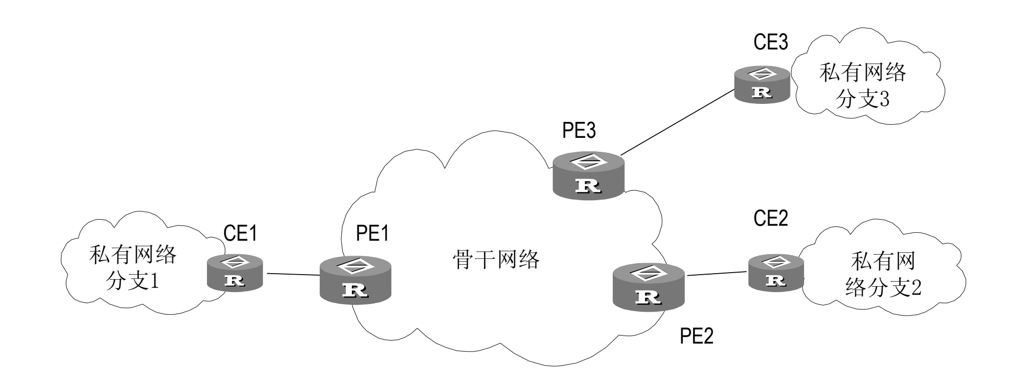


图1-6 基于 MPLS 的 VPN

图 1-6给出了基于MPLS的VPN的基本结构。CE（Customer Edge）是用户边缘设备，可以是路由器，也可以是交换机，甚至是一台主机；PE（Provider Edge）是服务商边缘路由器，位于骨干网络；PE负责对VPN用户进行管理、建立各PE间LSP连接、同一VPN用户各分支间路由分派。

PE 间的路由分派通常是用扩展的 BGP 协议实现的。基于 MPLS 的 VPN 支持不同分支间 IP 地址复用和不同 VPN 间互通，和传统的路由相比，VPN 路由中需要增加分支和 VPN 的标识信息，这就需要对 BGP 协议进行扩展才能携带 VPN 的路由信息。

1.5.2 基于 MPLS 的流量工程

1. 流量工程的作用

网络拥塞是影响骨干网络性能的主要问题。拥塞的原因一般是网络资源不足，或者网络资源的负载不均衡，导致局部拥塞。流量工程用来解决由负载不均衡导致的拥塞。流量工程通过动态监控网络的流量和网络单元的负载，实时调整流量管理参数、路由参数和资源约束参数等，使网络运行状态迁移到理想状态，优化网络资源的使用，从而避免由于负载不均衡引起的拥塞。

2. 用 MPLS 来实现流量工程的优点

现有的 IGP 协议都是拓扑驱动的，只考虑网络静态的连接情况，不能反映带宽和流量特性等动态状况，这正是导致网络负载不均衡的主要原因。而 MPLS 具有的一系列不同于 IGP 的特性，正是实现流量工程所需要的：MPLS 支持异于路由协议路径的显式 LSP 路由；LSP 较传统单个 IP 分组转发更便于管理和维护；基于约束路由的 LDP 可以实现流量工程的各种策略；基于 MPLS 的流量工程的系统开销较其它实现方式更低等等。

3. 基于 MPLS 的流量工程的实现

用 MPLS 来实现流量工程时，先要根据物理网络的拓扑生成 MPLS 导出图，即由 LSR、连接 LSR 的 LSP、LSP 属性这三种元素构成的派生拓扑图。同时，将通过

MPLS 域的数据划分成若干主干流（Traffic Trunk）。主干流一般定义为在 MPLS 域中经过相同 Ingress 和 Egress 的所有单向流量。主干流有许多属性，包括流量参数、路径选择和维护方式、优先级、可抢占性、资源亲和度等。对资源也定义了一些属性，如资源级别、最大分配复用度等。然后，以主干流的属性、资源的属性、网络的状态信息为生成约束路由的策略，来找出主干流的路径。主干流的路径可以根据网络状态的变化动态调整。

1.5.3 基于 MPLS 的 QoS

为了能够在 IP 网络上支持语音，视频等实时业务，需要有 QoS 的支持，以便保证重要的、敏感或者实时性较强的数据流在网络中得到优先处理。华为设备支持基于 MPLS 的流量工程和 Diff-serv 特性，在保证网络高效利用率的同时，又能根据不同数据流的优先级实现差别服务，从而为语音，视频数据流提供有带宽保证的、低延时、低丢包率的服务。由于目前全网实施流量工程的难度比较大，因此，在实际的组网方案中，往往倾向于使用差分服务模型来实施 QoS。

Diff-Serv 的基本机制是：在网络边缘，根据业务的服务质量要求，将该业务映射到一定的业务类别中，利用 IP 分组中的 DS 字段（由 TOS 域而来）唯一的标记该类业务，然后，骨干网络中的各节点根据该字段对各种业务采取预先设定的服务策略，保证相应的服务质量。Diff-Serv 的这种对服务质量的分类和标签机制和 MPLS 的标签分配十分相似，事实上，基于 MPLS 的 Diff-Serv 就是通过将 DS 的分配融入 MPLS 的标签分配过程来实现的。

Diff-Serv 对不同的服务类别规定了一致的处理方法，包括队列选择、排队、丢弃等操作，这些处理组合就叫 PHB（Per Hop Behavior）。同时，属于同一 PHB 的分组又可以有不同的丢弃优先级。PHB 和丢弃优先级信息通过为分组分配不同的 DS 编码来表示，这些 DS 编码又称 DSCP（Diff-Serv Code Point）。关于 Diff-Serv 的详细介绍，请参见本手册的 QoS 配置部分。

为了支持基于 Diff-Serv 模型的端到端的 QoS 服务，华为提供如下几种技术手段：

- 基于流量的 IP 优先级分类

IP 优先级分类在网络边缘进行，利用 IPv4 包头的 Type-of-Service 3 个比特对每一个 IP 包依据其地址进行优先级分类。在核心利用不同的队列技术对不同等级的流量进行不同的处理，使得不同的服务级别得到体现。为实现语音、图象、数据数据流的差分服务，对不同的业务流在进行标签交换时，即 PE 在给报文加 Label 时，会把 IP 报文携带的 TOS 值映射到标签的 CoS 域，这样，原来由 IP 携带的类型信息，现在由标签携带。在 PE 路由器之间，根据标签的 CoS 域，进行有差别的调度（PQ、CQ、WFQ、CBQ 等）。

- 用 TP 实现承诺带宽及限制带宽的功能

在 PE 上与 CE 相连的链路上配置 TP（Traffic Policing）可以实现该功能。同时，TP 还提供了承诺的带宽和限制带宽的功能。

- 用 WRED 进行拥塞避免

WRED 在网络的瓶颈处监视并缓解网络的拥塞。一般在接入层出现拥塞的概率比较大。WRED 监视网络的负载，当拥塞刚开始出现时，它就开始有选择地丢弃一些包以降低流量。WRED 丢包的策略为：低优先级的流先丢，以保证高优先级的流可以顺畅通过。在可能发生拥塞的端口运行 WRED，是避免拥塞的较好选择。

在具体实现中，为了达到最好的效率，需要对任务进行分工。因为 QoS 是一个需要消耗很多处理器资源的应用，所以，这一任务应分配在边缘和核心路由器上运行，以减少对单独路由器的压力。

综上所述，实现基于 CoS 的差分服务结构需要 4 个步骤：

- MPLS 边缘路由器上实现入口的带宽限制完成入口流量的分类。
- 边缘设备也需要承担带宽管理的工作，采用 TP。
- MPLS 核心路由器完成 CoS 的管理工作，进行有差别的服务质量保证。
- 出口设备，像入口设备一样，完成带宽限制工作。入口、出口设备对带宽的限制保护了网络免于拥塞，使得网络具有很高的可扩展性。

详细内容请参见 QoS 部分。

第2章 MPLS 基本能力配置

2.1 MPLS 的基本能力配置概述

MPLS 提供了比较完全的 MPLS 基本能力：

- 基本 MPLS 转发

VRP 支持基本 MPLS 转发功能。除标签报文的转发之外，还提供了 TTL 处理等功能。

- LDP 会话建立和 LSP 路径维护

支持 LDP 会话；支持最大跳数和路径向量两种方式的环路检测；提供静态 LSP 的建立、删除功能。

支持松散和严格的显式路由；可以人工指定 LSP 路径。

VRP 除提供 MPLS 基本功能外，还提供了性能监视和故障检测工具。

要使一台路由器具有基本的 MPLS 功能，一般的配置过程如下：

(1) 配置 LSR 的标识 ID

(2) 使能 MPLS

(3) 使能 LDP 协议

(4) 进入接口模式，使能接口的 LDP 功能。

经过上述的基本配置，路由器即可提供 MPLS 转发和 LDP 信令功能。

如果要修改一些缺省参数，或者实现一些特殊的 MPLS 功能，如手工建立 LSP、建立显式路由等，则可以根据配置列表提供的方法来配置。有些复杂的功能，可能需要多个配置的组合才能实现。

2.2 MPLS 基本配置

MPLS 基本配置中必须的配置项包括：

- 配置 MPLS LSR ID
- 使能 MPLS，并进入 MPLS 视图

MPLS 基本配置中可选的配置项包括：

- 配置拓扑驱动建立 LSP 的建立策略
- 配置静态 LSP
- 配置上报统计信息的间隔时间

2.2.1 配置 MPLS LSR ID

在配置其他 MPLS 命令之前，必须首先为 LSR 配置 ID。LSR ID 一般采用 IP 地址的格式，并且要保证域内唯一。

请在系统视图下进行下列配置。

表2-1 配置 MPLS LSR ID

操作	命令
指定 LSR 的 LSR ID	**mpls lsr-id** *ip-address*
删除 LSR 的 LSR ID	**undo mpls lsr-id**

缺省未指定 LSR ID。

2.2.2 进入 MPLS 视图

在系统视图下，**mpls** 命令用来进入 MPLS 视图。在进入 MPLS 视图后，可以进行 MPLS 相关配置。

表2-2 进入 MPLS 视图

操作	命令
进入 MPLS 视图	**mpls**

然后，在接口视图下，**mpls** 命令用来使能接口的 MPLS 能力。

在不支持广播报文的链路层协议，如 X.25、帧中继、ATM 上，必须要使用命令 **protocol ip** { *ip-address* [*ip-mask*] | **default** | **inarp** [*minutes*] } [**broadcast**] 配置 **broadcast** 属性，以支持广播和组播报文的传递。

2.2.3 配置拓扑驱动建立 LSP 的建立策略

配置拓扑驱动建立 LSP 的建立策略，指定过滤策略为 all 和 ip-prefix 策略。

请在 MPLS 视图下进行下列配置。

表2-3 配置拓扑驱动建立 LSP 的建立策略

操作	命令
配置拓扑驱动建立 LSP 的建立策略	**lsp-trigger { all | ip-prefix** *ip-prefix* **}**
取消参数所指定的过滤条件，任何路由都不允许触发建立 LSP	**undo lsp-trigger** { **all** | **ip-prefix** *ip-prefix* }

2.2.4 配置静态 LSP

可以手工设置某 LSR 为一条 LSP 上的一个节点，并可以对该 LSP 上承载的数据流进行限制。根据在 MPLS 域中的不同位置，LSR 有三种节点情况：入节点（Ingress）、中间节点（Transit）、出节点（Egress）。值得注意的是，必须对指定 LSP 沿途的 LSR 均作了相应配置，这条 LSP 才能正常工作。

undo static-lsp 用于删除一个用手工方式建立的 LSP。

请在 MPLS 视图下进行下列配置。

表2-4 设置本 LSR 为指定 LSP 的节点

操作	命令
设置本 LSR 为指定 LSP 的入口节点	**static-lsp ingress** *lsp-name* { **destination** *dest-addr* { *addr-mask* | *mask-length* } | **l2vpn** } { **nexthop** *next-hop-addr* | **outgoing-interface** { *interface-name* | *interfac-type interfac-num* } } **out-label** *out-label-value*
	undo static-lsp ingress *lsp-name* [**l2vpn**]
设置本 LSR 为指定 LSP 的中间节点	**static-lsp transit** *lsp-name* [**l2vpn**] **incoming-interface** *interface-type interface-num* } **in-label** *in-label-value* { { **nexthop** *next-hop-addr* } | { **outgoing-interface** { *interface-name* | { *interface-type interface-num* } } } } **out-label** *out-label-value*
	undo static-lsp transit *lsp-name* [**l2vpn**]
设置本 LSR 为指定 LSP 的出口节点	**static-lsp egress** *lsp-name* [**l2vpn**] **incoming-interface** { *interface-name* | *interfac-type interfac-num* } **in-label** *in-label-value*
	undo static-lsp egress *lsp-name* [**l2vpn**]

2.2.5 配置上报统计信息的间隔时间

请在 MPLS 视图下进行下列配置。

表2-5 配置上报统计信息的间隔时间

操作	命令
配置上报统计信息的间隔时间	**statistics interval** *interval-time*
恢复缺省值	**undo statistics interval**

缺省情况是间隔为 0，即不上报。

2.3 LDP 协议配置

LDP 协议配置中必须的配置包括：

- 使能 LDP 协议

- 在接口上使能 LDP 协议

LDP 协议配置中可选的配置包括：

- 配置 LDP 扩展发现模式
- 在出口节点配置倒数第二跳的标签
- 配置会话参数
- 配置环路检测
- 配置 LDP 的验证方式

2.3.1 使能 LDP 协议

要进行 LDP 的配置，首先要使能 LDP 协议。

请在系统视图下进行下列配置。

表2-6 使能/去使能 LDP 协议

操作	命令
使能 LDP 协议	**mpls ldp**
禁止 LDP 协议	**undo mpls ldp**

缺省情况下 LDP 协议未激活。

2.3.2 在接口上使能 LDP 协议

要使接口具有 LDP 功能，必须在该接口模式下，对接口进行 LDP 使能。使能后的接口即开始建立 LDP 会话，在拓扑驱动方式时，开始建立 LSP。

禁止端口 LDP 功能，会导致接口下的所有 LDP 会话中断，基于这些会话的所有 LSP 也将被删除，建议用户谨慎使用此命令。

请在接口视图下进行下列配置。

表2-7 接口 LDP 使能

操作	命令
允许接口 LDP 功能	**mpls ldp enable**
禁止接口 LDP 功能	**mpls ldp disable**

缺省情况下禁止接口的 LDP 功能。

2.3.3 配置 LDP 扩展发现模式

配置 remote-peer，主要用于扩展发现模式，以便与链路不直接相连的对等体建立会话。

1. 进入 remote-peer 模式

请在系统视图下进行下列配置。

表2-8 进入扩展发现模式

操作	命令
进入扩展发现模式	**mpls ldp remote-peer** *index*
删除相应的 remote-peer	**undo mpls ldp remote-peer** *index*

无缺省的 remote-peer。

2. 配置 remote-peer 的地址

可以指定 remote-peer 的任何一个使能了 LDP 的接口地址或发布了路由的 LSR 的 loopback 地址，作为 remote-peer 的地址。

请在 remote-peer 视图下进行下列配置。

表2-9 配置 remote-peer 的地址

操作	命令
配置 remote-peer 的地址	**remote-ip** *ip-address*

无缺省的 remote-peer。

2.3.4 在出口节点配置倒数第二跳的标签

在出口节点处，指明倒数第二跳分发怎样的标签。

请在 MPLS 视图下进行下列配置。

表2-10 在出口节点配置倒数第二跳的标签

操作	命令
出口节点处指明倒数第二跳分发怎样的标签	**mpls label advertise** { **implicit-null** | **explicit-null** | **non-null** }
恢复缺省值	**undo mpls label advertise** { **implicit-null** | **explicit-null** | **non-null** }

explicit-null-label：在出口处指定向倒数第二跳分配显式空标签，即。

implicit-null-label：在出口处指定向倒数第二跳分配隐式空标签。

non-null-label：在出口处指定向倒数第二跳分配普通标签。

如果倒数第二跳分配了显式空标签，则显示空标签只能位于标签栈底。

缺省为隐式标签。

2.3.5 配置会话参数

1. 配置会话保持时间

接口上的 LDP 实体周期性的发出 Hello 报文，来发现 LDP 对等体，已经建立的 LDP 会话，也必须靠定期的消息来维持其存在（如果没有 LDP 消息报文，则须发送 Keepalive 报文）。

 注意：

修改 holdtime 参数，会导致原有的会话重新建立，原有的建立在此会话上的 LSP 也会被删除、重建。这里的会话指的是链路会话，而非 remote 会话。

请在接口视图下进行下列配置。

表2-11 配置会话保持时间

操作	命令
配置会话保持时间	**mpls ldp timer** { **session-hold** *session-holdtime* | **hello** *hello-holdtime* }
恢复会话保持时间为缺省值	**undo mpls ldp timer** { **session-hold** | **hello** }

session-holdtime 和 *hello-holdtime* 的缺省值为 15 秒。

📖 说明：

对于 ATM，接口相关配置命令只支持在 point-to-point 模式的 ATM 子接口下进行配置。

在不支持广播报文的链路层协议，如帧中继、ATM 上，必须要配置 broadcast 属性，以支持广播和组播报文的传递。

2. 配置 hello 消息中的传输地址

这里的传输地址就是 hello 消息的传输地址 TLV 中携带的地址。通常情况下，传输地址为本 LSR 的 MPLS LSR ID，但是，有些应用需要灵活的配置。

请在接口视图下进行下列配置。

表2-12 配置 hello 传输地址

操作	命令
配置 hello 传输地址	**mpls ldp transport-ip** { **interface** | *ip-address* }
恢复 hello 传输地址为默认值	**undo mpls ldp transport-ip**

缺省传输地址为本 LSR 的 MPLS LSR ID。

当两个 LSR 邻居之间有多条链路直接相连并且都使能 mpls ldp 时，要求使能 mpls ldp 的多个链路必须配置相同的传输地址，建议采用缺省的 lsr-id 作为传输地址，否则将可能导致 ldp 会话无法稳定建立。

2.3.6 配置环路检测

1. 环路检测使能

用于控制在 LDP 信令过程中是否使用环路检测功能，环路检测有最大跳数和路径向量两种方式。

最大跳数方式是在传递标签绑定的消息中包含跳数信息，每经过一跳，该值就加一，当该值超过规定的最大值时，就认为出现了环路，从而终止 LSP 的建立过程。

路径向量方式是在传递标签绑定的消息中记录路径信息，每经过一跳，相应的路由器就检查自己的 ID 是否在此记录中，如果没有，就将自己的 ID 添加到该记录中，若有，就说明出现了环路，终止 LSP 的建立过程。

请在系统视图下进行下列配置。

表2-13 环路检测使能

操作	命令
允许进行环路检测	**mpls ldp loop-detect**
禁止进行环路检测	**undo mpls ldp loop-detect**

缺省为不允许环路检测。

2. 设置环路检测最大跳数

当环路检测采用最大跳数方式时，可以规定跳数的最大值，超过该最大值即认为出现了环路，LSP 建立失败。

请在系统视图下进行下列配置。

表2-14 设置环路检测最大跳数

操作	命令
设置环路检测的最大跳数	**mpls ldp hops-count** *hop-number*
恢复最大跳数的缺省值	**undo mpls ldp hops-count**

缺省的最大值为 32。

3. 设置路径向量的最大值

当环路检测采用路径向量方式时，也需要规定 LSP 路径的最大值。这样，在以下条件之一时，即认为出现了环路，LSP 建立失败：

(1) 路径向量记录表中已有本 LSR 的记录

(2) 路径的跳数超过这里设定的最大值

请在系统视图下进行下列配置。

表2-15 设置路径向量的最大值

操作	命令
设置路径向量的最大跳数	**mpls ldp path-vectors** *pv-number*
恢复路径向量最大跳数的缺省值	**undo mpls ldp path-vectors**

路径向量的最大跳数缺省的最大值为 32。

2.3.7 配置 LDP 验证方式

请在接口视图或 remote-peer 视图下进行下列配置。

表2-16 配置 LDP 验证方式

操作	命令
配置 LDP 验证方式	**mpls ldp password** [**cipher** | **simple**] *password*
取消 LDP 的验证。	**undo mpls ldp password**

2.4 MPLS 基本能力显示与调试

2.4.1 MPLS 的显示与调试

MPLS 提供了丰富的显示与调试命令，可以监控 LDP 会话状态、隧道配置情况、所有 LSP 及其状态等，是调试、诊断的有力工具。

1. 显示静态 LSP

在完成上述配置后，在所有视图下，执行 **display** 命令，可以显示全部或单个静态 LSP 的运行情况，通过查看显示信息，验证配置的效果。

表2-17 显示静态 LSP 的信息

操作	命令
显示静态 LSP 的信息	**display mpls static-lsp** { **include** | **verbose** }

2. 显示/清除 MPLS 统计数据

在完成上述配置后，在所有视图下，执行 **display** 命令，可以显示全部或单个 LSP 的统计信息，通过查看显示信息，验证配置的效果。

表2-18 显示/清除 MPLS 统计数据

操作	命令
显示 MPLS 统计数据	**display mpls statistics** { **interface** { **all** | *interface-typeinterface-type interface-num* } } | { **lsp** [*lsp-Indexname* | **all** | **name**] } }
清除 MPLS 统计数据	**reset mpls statistics** { **interface** { **all** | *interface-type interface-num* } | **lsp** {*lsp-index* | **all** | **name** *lsp-name* } }

3. 显示使能 MPLS 的接口信息

在完成上述配置后，在所有视图下执行 **display** 命令，可以显示所有使能了 MPLS 能力的接口的相关信息，通过查看显示信息，验证配置的效果。

表2-19 显示使能了 MPLS 的接口信息

操作	命令
显示使能 MPLS 的接口信息	**display mpls interfaces**

4. 显示 LSP 的信息

在所有视图下执行下列命令，可以显示 MPLS LSP 的相关信息。

表2-20 显示 MPLS LSP 的相关信息

操作	命令
显示 LSP 的信息。	**display mpls lsp {include text | verbose }**

5. MPLS 的调试信息

在用户视图下，执行 **debugging** 命令，可对所有使能了 MPLS 能力的接口的相关信息进行调试。

打开调试开关对路由器性能有一定影响，建议慎用此命令。该命令的 **undo** 形式关闭相应的调试开关。

表2-21 打开 MPLS 的调试信息开关

操作	命令
打开 MPLS 的各种调试信息的开关	**debugging mpls lspm** { **all** | **packet** | **event** | **process** | **agent** | **interface** | **policy** | **vpn** }
关闭 MPLS 的各种调试信息的开关	**undo debugging mpls lspm** { **all** | **packet** | **event** | **process** | **agent** | **interface** | **policy** | **vpn** }
复位接口上某一个指定的 LDP 会话。	**mpls ldp reset-session** *peer-address*

6. MPLS 的 trap 信息

使能 MPLS 的 LSP/LDP 建立过程的 TRAP 功能。

请在系统视图下进行操作。

表2-22 使能 MPLS 的 TRAP 功能

操作	命令
使能 MPLS 的 LDP TRAP 功能	**snmp-agent trap enable ldp**
去使能 MPLS 的 LDP TRAP 功能	**undo snmp-agent trap enable ldp**
使能 MPLS 的 LSP TRAP 功能	**snmp-agent trap enable lsp**
去使能 MPLS 的 LSP TRAP 功能	**undo snmp-agent trap enable lsp**

2.4.2 LDP 协议的显示与调试

1. LDP 协议的显示命令

VRP 提供了丰富的 MPLS 监控操作命令，可以监控 LSR 状态、LDP 会话状态、接口状态、对等体状态等，是调试、诊断的有力工具。

在完成上述配置后，在所有视图下，执行 **display** 命令，可以显示配置后 LDP 的运行情况，通过查看显示信息，验证配置的效果。

表2-23 LDP 监控命令

操作	命令
显示 LD 协议信息	**display mpls ldp**
显示 LDP 的 **buffer** 信息。	**display mpls ldp buffer-info**
显示 LDP 使能的接口信息	**display mpls ldp interface**
显示 LDP 保存的标签信息	**display mpls ldp lsp**
显示 LDP 会话的所有对等体信息	**display mpls ldp peer**
显示 LDP 会话的远端对等体信息	**display mpls ldp remote**
显示 LDP 内存池信息	**display mpls ldp pool**
显示 LDP 会话状态和参数	**display mpls ldp session**

2. 诊断命令

在用户视图下，执行 **debugging** 命令，可对 LDP 的各种消息进行调试。

表2-24 MPLS 诊断

操作	命令
打开 MPLS 诊断开关	**debugging mpls ldp** { **all** | **main** | **advertisement** | **session** | **pdu** | **notification** | **remote** }
关闭 MPLS 诊断开关	**undo debugging mpls ldp** { **all** | **main** | **advertisement** | **session** | **pdu** | **notification** | **remote** } [**interface** *interface-type interface-num*]

all：显示所有和 LDP 相关的调试信息。

main：显示 LDP 主任务的调试信息。

advertisement：显示处理 LDP 通告消息过程中的调试信息。

session：显示处理 LDP 会话过程中的调试信息。

pdu：显示处理 PDU 数据包过程中的调试信息。

notification：显示处理通知消息过程中的调试信息。

remote：显示所有 remote peer 的调试信息。

2.5 MPLS 基本能力典型配置举例

1. 组网需求

图 2-1给出一个由四台Router路由器组成的网络，其中RouterB和RouterC间通过SDH连接，RouterB和RouterA、RouterD间通过以太网连接。

四台路由器均支持 MPLS，任意路由器之间都可以建立 LSP，运行的路由协议为OSPF。LDP 利用 OSPF 的路由信息建立 LSP。

2. 组网图

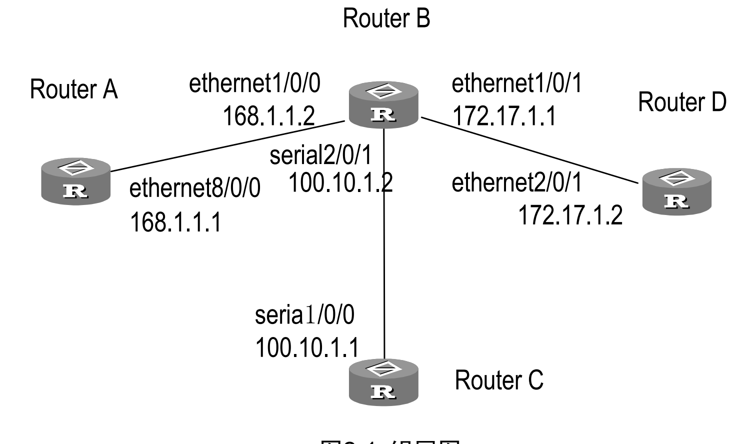


图2-1 组网图

3. 配置步骤

(1) RouterA 上的配置

配置 RouterA 的 LSR ID，并使能 MPLS 及 LDP

```
[Quidway] mpls lsr-id 168.1.1.1
[Quidway] mpls
[Quidway] mpls ldp
```

配置以太网接口的 IP 地址，并对接口使能 MPLS 及 LDP

```
[Quidway] interface ethernet 8/0/0
[Quidway-Ethernet8/0/0] ip address 168.1.1.1 255.255.0.0
[Quidway-Ethernet8/0/0] MPLS
[Quidway-Ethernet8/0/0] mpls ldp enable
```

在 RouterA 与 RouterB 相连的接口上运行 OSPF

```
[Quidway] router id 168.1.1.1
[Quidway] ospf
[Quidway-ospf-1] area 0
[Quidway-ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
```

(2) Router B 上的配置

配置 Router B 的 LSR ID，并使能 MPLS 及 LDP

```
[Quidway] mpls lsr-id 172.17.1.1
[Quidway] mpls
[Quidway] mpls ldp
```

配置以太网接口 1/0/0 的 IP 地址，并对接口使能 MPLS 及 LDP

```
[Quidway] interface ethernet 1/0/0
[Quidway-Ethernet1/0/0] ip address 168.1.1.2 255.255.0.0
[Quidway-Ethernet1/0/0] mpls
```

置以太网接口 1/0/1，并对接口使能 MPLS 及 LDP

```
[Quidway] interface ethernet 1/0/1
[Quidway-Ethernet1/0/1] ip address 172.17.1.1 255.255.0.0
[Quidway-Ethernet1/0/1] mpls
[Quidway-Ethernet1/0/1] mpls ldp enable
```

置 SERIAL 接口 2/0/1，并对接口使能 MPLS 及 LDP

```
[Quidway] interface serial 2/0/1
[Quidway-Serial2/0/1] ip address 100.10.1.2 255.255.255.0
[Quidway-Serial2/0/1] mpls
[Quidway-Serial2/0/1] mpls ldp enable
[Quidway-Serial2/0/1] quit
```

分别在 RouterB 与 RouterA、RouterD、RouterC 相连的接口上运行 OSPF

```
[Quidway] router id 172.17.1.1
[Quidway] ospf
[Quidway-ospf-1] area 0
[Quidway-ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 172.17.0.0 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 100.10.1.0 0.0.0.255
[Quidway-ospf-1-area-0.0.0.0] quit
```

(3) Router C 上的配置

配置 Router C 的 LSR ID，并使能 MPLS 及 LDP

```
[Quidway] mpls lsr-id 172.16.1.2
[Quidway] mpls
[Quidway] mpls ldp
```

配置 SERIAL 接口 1/0/0 的 IP 地址，并对接口使能 MPLS 及 LDP

```
[Quidway] interface serial 1/0/0
[Quidway-Serial1/0/0] ip address 100.10.1.1 255.255.255.0
[Quidway-Serial1/0/0] mpls
[Quidway-Serial1/0/0] mpls ldp enable
```

```
[Quidway-Serial1/0/0] quit
```

在 RouterC 与 RouterB 相连的接口上运行 OSPF

```
[Quidway] router id 172.16.1.2
[Quidway] ospf
[Quidway-ospf-1] area 0
[Quidway-ospf-1-area-0.0.0.0] network 100.10.1.0 0.0.0.255
```

(4) Router D 上的配置

配置 RouterD 的 LSR ID，并使能 MPLS 及 LDP

```
[Quidway] mpls lsr-id 172.17.1.2
[Quidway] mpls
[Quidway] mpls ldp
```

配置以太网接口 2/0/1 的 IP 地址，并对接口使能 MPLS 及 LDP

```
[Quidway] interface ethernet 2/0/1
[Quidway-Ethernet2/0/1] ip address 172.17.1.2 255.255.0.0
[Quidway-Ethernet2/0/1] mpls
[Quidway-Ethernet2/0/1] mpls ldp enable
```

在 RouterD 与 RouterB 相连的接口上运行 OSPF

```
[Quidway] router id 172.17.1.2
[Quidway] ospf
[Quidway-ospf-1] area 0
[Quidway-ospf-1-area-0.0.0.0] network 172.17.0.0 0.0.255.255
```

2.6 MPLS 配置的故障排除

故障现象：接口使能 LDP 协议后，不能与对端建立会话。

故障排除：

原因之一：环路检测配置不同。

解决方法：检查本地和对端的配置，是否一端配了环路检测，而另一端没有配，如果是这样，就会导致会话协商不通过。

原因之二：如果本机得不到对方的 LSR ID 的路由，就不能建立 TCP 连接，会话就不可能建立。

解决方法：建立会话时默认的传输地址是 MPLS LSR ID，本机必须将 LSR ID 的路由（一般为 loopback 地址）发布出去，同时，要学到对端的 LSR ID 的路由。

第3章 BGP/MPLS VPN 配置

3.1 BGP/MPLS VPN 概述

传统 VPN 使用第二层隧道协议（L2TP、L2F 和 PPTP 等）或者第三层隧道技术（IPSec、GRE 等），获得了很大成功，被广泛应用。但是，随着 VPN 范围的扩大，传统 VPN 在可扩展性和可管理性等方面的缺陷越来越突出；另外，QoS（Quality of Service—服务质量）和安全问题也是传统 VPN 难以解决的问题。

通过 MPLS（Multiprotocol Label Switching，多协议标签交换）技术可以非常容易地实现基于 IP 技术的 VPN 业务，而且可以满足 VPN 可扩展性和管理的需求。利用 MPLS 构造的 VPN，还提供了实现增值业务的可能；通过配置，可将单一接入点形成多种 VPN，每种 VPN 代表不同的业务，使网络能以灵活方式传送不同类型的业务。

VRP 目前提供比较完全的 BGP/MPLS VPN 组网能力：

- 地址隔离，允许不同 VPN 和公网的地址重叠。
- 支持 MP-BGP 协议穿越公网发布 VPN 的路由消息，构建 BGP/MPLS VPN。
- 通过 MPLS LSP 转发 VPN 的数据流。
- 提供了 MPLS VPN 的性能监视和故障检测工具。

3.1.1 BGP/MPLS VPN 模型

1. BGP/MPLS VPN 模型

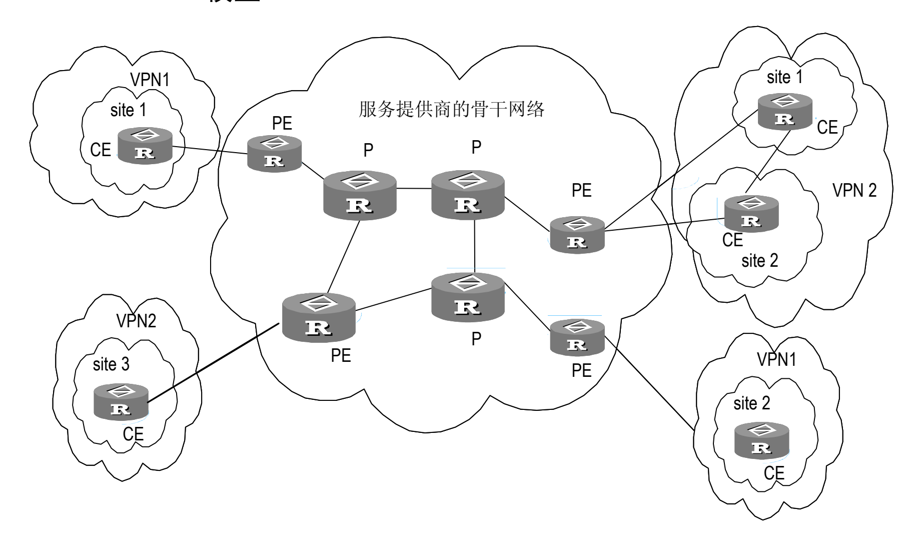


图3-1 MPLS VPN 模型

如上图所示，MPLS VPN 模型中，包含三个组成部分：CE、PE 和 P。

- CE（Customer Edge）设备：是用户网络边缘设备，有接口直接与服务提供商相连，可以是路由器或是交换机等。CE“感知”不到 VPN 的存在。
- PE（Provider Edge）路由器：即运营商边缘路由器，是运营商网络的边缘设备，与用户的 CE 直接相连。MPLS 网络中，对 VPN 的所有处理都发生在 PE 路由器上。
- P（Provider）路由器：运营商网络中的骨干路由器，不和 CE 直接相连。P 路由器需要支持 MPLS 能力。

CE 和 PE 的划分主要是从运营商与用户的管理范围来划分的，CE 和 PE 是两者管理范围的边界。

2. BGP/MPLS VPN 中的基本概念

(1) vpn-instance

vpn-instance 是 MPLS VPN 中实现 VPN 路由的重要概念。在 MPLS VPN 的实现中，每个 Site 在 PE 上对应一个专门的 vpn –instance（vpn –instance 通过与接口绑定实现与 Site 的关联）。如果一个 Site 中的用户同时属于多个 VPN，则该 Site 对应的 vpn-instance 中将包括所有这些 VPN 的信息。

具体来说，vpn-instance 的信息中包括：标签转发表、IP 路由表、与 vpn-instance 绑定的接口以及 vpn-instance 的管理信息（包括 RD、路由过滤策略 VPN Target、成员接口列表等）。可以认为，它综合了该 Site 的 VPN 成员关系和路由规则。

PE 负责更新和维护 vpn-instance 与 VPN 的关联关系。为了避免数据泄漏出 VPN 之外，同时防止 VPN 之外的数据进入，在 PE 上，每个 vpn-instance 有一套相对独立的路由表和标签转发表，报文转发信息存储在该 vpn-instance 的 IP 路由表和标签转发表中。

(2) MP-BGP

MP-BGP（multiprotocol extensions for BGP-4，请参见 RFC2283）在 PE 路由器之间传播 VPN 组成信息和路由。MP-BGP 向下兼容，既可以支持传统的 IPv4 地址族，又可以支持其他地址族（比如 VPN-IPv4 地址族）。使用 MP-BGP 确保 VPN 的私网路由只在 VPN 内发布，并实现 MPLS VPN 成员间的通信。

(3) VPN-IPv4 地址族

由于 VPN 网络是一个私用网络，可以使用相同的 IP 地址来表示不同的 Site。而 PE 路由器之间使用 MP-IBGP 来发布与之相连的 CE 的路由时，是假定 IP 地址是全球唯一的，二者之间不同的含义会导致路由错误。为了解决这个问题，在发布路由之前 MP-BGP 需要实现 IPv4 地址到 VPN-IPv4 地址族的转换，使之成为全球唯一的地址（故 PE 路由器需要支持 MP-BGP）。

一个 VPN-IPv4 地址有 12 个字节，开始是 8 字节的 RD（Route Distinguisher，路由分辨符），下面是 4 字节的 IPv4 地址。服务供应商可以独立地分配 RD，但是，需要把他们专用的 AS（Autonomous System—自治系统）号作为 RD 的一部分。通过这样的处理以后，即使 VPN-IPv4 地址中包含的 4 字节 IPv4 地址重叠，VPN-IPv4 地址仍可以保持全局唯一。RD 纯粹是为了区别不同的路由，仅在运营商网络内部使用，RD 为零的 VPN-IPv4 地址相当于普通的 IPv4 地址。

PE 从 CE 接收的路由是 IPv4 路由，需要引入 vpn-instance 路由表中，此时需要附加一个 RD。在我们的实现中，为来自于同一个用户 Site 的所有路由设置相同的 RD。

3. VPN Target 属性

VPN Target 属性是 MP-BGP 扩展团体属性之一，主要用来限制 VPN 路由信息发布。它标识了可以使用某路由的 Site 的集合，即该路由可以被哪些 Site 所接收，通过它，可以明确每一个 PE 路由器可以接收哪些 Site 传送来的路由。与 VPN Target 中指明的 Site 相连的 PE 路由器，都会接收到具有这种属性的路由。

PE 路由器存在两个 VPN Target 属性集合：一个集合称为 Export Targets，在发布本地路由到远端 PE 路由器时，附加到从某个直连的 Site 上接收到的路由上；另一个集合称为 Import Targets，在接收远端 PE 发布的路有时，决定哪些路由可以引入此 Site 的 VPN 路由表中。

当通过匹配路由所携带的 VPN Target 属性来过滤 PE 路由器接收的路由信息时，如果 Export VPN Target 集合与 Import VPN Target 集合存在相同项，则该路由被安装到 VPN 路由表中，进而发布给相连的 CE；如果 Export VPN Target 集合与 Import VPN Target 集合没有相同项，则该路由被拒绝。

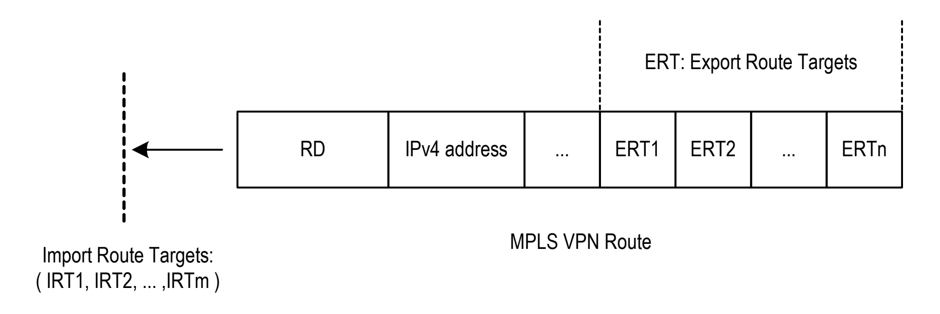


图3-2 通过匹配 VPN Target 属性过滤路由

📖 说明：

通过 VPN Target 属性来过滤 PE 路由器接收的路由信息，使得其他 VPN 的路由不会出现在本 VPN 路由表中，所以，CE 发送的数据只会在所属的 VPN 内转发，不会被转发到外部。

3.1.2 BGP/MPLS VPN 的实现

BGP/MPLS VPN 的主要原理是：利用 BGP 在运营商骨干网上传播 VPN 的私网路由信息，用 MPLS 来转发 VPN 业务流。

下面从 VPN 路由信息的发布和 VPN 报文转发两个方面介绍 BGP/MPLS VPN 的实现。

1. VPN 路由信息发布

(1) CE 到 PE 间的路由信息交换

PE 可以通过静态路由、RIP（应支持多实例）、OSPF（应支持多实例）或 EBGP 学习到与它相连的 CE 的路由信息，并将此路由安装到 VPN-instance 中。

(2) 入口 PE 到出口 PE 的路由信息交换

入口 PE 路由器利用 MP-IBGP 穿越公网，把它从 CE 学习到的路由信息发布给出口 PE（带着 MPLS 标签），同时，获得出口 PE 学习到的 CE 路由信息，。

PE 之间通过 IGP（如 RIP、OSPF）来保证 VPN 内部节点之间的连通性，故应在所有互联接口及 loopback 接口上运行 IGP。

(3) PE 之间的 LSP 建立

为了使用 MPLS LSP 转发 VPN 的数据流量，一定在 PE 之间建立 LSP。从 CE 接收报文并建立标签栈的 PE 路由器是 ingress LSR，BGP 的下一跳（即出口 PE 路由器）是 engress LSR。可以使用 LDP 建立尽力转发的 LSP，也可以使用 RSVP 建立支持特定 QoS 的 LSP 或基于流量工程的 LSP。使用 LDP 建立 LSP 将在 PE 之间形成全连接的 LSP（本章只讨论前者）。

(4) PE 到 CE 间的路由信息交换

CE 可以通过静态路由、RIP、OSPF、或 EBGP，从相连的 PE 上学习远端的 VPN 路由。

经过以上的步骤，CE 之间将建立可达的路由，完成 VPN 私网路由信息在公网上的传播。

2. VPN 报文的转发

VPN 报文在入口 PE 路由器上形成两层标签栈：

内层标签，也称 MPLS 标签，是由出口 PE 向入口 PE 发布路由时分配的（安装在 VPN 转发表中），在标签栈中处于栈底位置。当从公网上发来的 VPN 报文从 PE 到达 CE 时，根据标签查找 MPLS 转发表就可以从指定的接口将报文发送到指定的 CE 或者 Site。

外层标签，也称 LSP 的初始化标签，指示了从入口 PE 到出口 PE 的一条 LSP，在标签栈中处于栈顶位置。VPN 报文利用这层标签的交换，就可以沿着 LSP 到达对端 PE。

以下图为例：

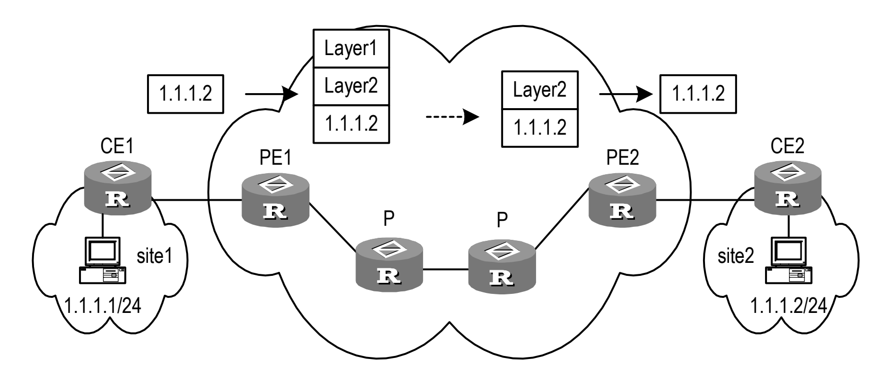


图3-3 VPN 报文转发示意图

(1) Site1 发出一个目的地址为 1.1.1.2 的 IPv4 报文到达 CE1，CE1 查找 IP 路由表，根据匹配的表项将 IPv4 报文发送至 PE1。

(2) PE1 根据报文到达的接口及目的地址查找 VPN-instance 表项，获得内层标签、外层标签、BGP 下一跳（PE2）、输出接口等。建立完标签栈后，PE1 通过输出接口转发 MPLS 报文到 LSP 上的第一个 P。

(3) LSP 上的每一个 P 路由器利用交换报文的外层标签转发 MPLS 报文，直到报文传送到倒数第二跳路由器，即到 PE2 前的 P 路由器。倒数第二跳路由器将外层标签弹栈，并转发 MPLS 报文到 PE2。

(4) PE2 根据内层标签和目的地址查找 VPN 转发表，确定标签操作和报文的出接口，最终弹出内层标签并由出接口转发 IPv4 报文至 CE2。

(5) CE2 查找路由表，根据正常的 IPv4 报文转发过程将报文传送到 Site2。

3.1.3 分层式 BGP/MPLS VPN 的实现

在 BGP/MPLS VPN 中，由于 PE 设备要汇聚多个 VPN 路由，在大规模部署时或者 PE 设备容量较小时容易形成瓶颈。为解决这一问题，华为公司提出了 HoVPN（Hiberarchy of VPN，分层式 BGP/MPLS VPN）的解决方案。

分层式 BGP/MPLS VPN 将一个 MPLS VPN 网络划分成多个 MPLS VPN 网络，且形成一个层次化的 BGP/MPLS VPN 网络结构，各层 VPN 网络承担不同的角色，处于网络较高层次的 MPLS VPN 网络的 PE 设备的路由和转发性能要求比较高，主要实现骨干网互连和大的 VPN 网络客户的接入功能，较低层次的 MPLS VPN 网络的 PE 设备的路由和转发性能要求较低，主要实现边缘的 VPN 客户接入。该种网络模型同典型的 IP 网络模型相吻合，提高了 BGP/MPLS VPN 的可扩展性，可以实现在

低端设备组成的下层 MPLS VPN 网络上做 MPLS VPN 接入，然后通过高端的 MPLS VPN 网络骨干网络互连。

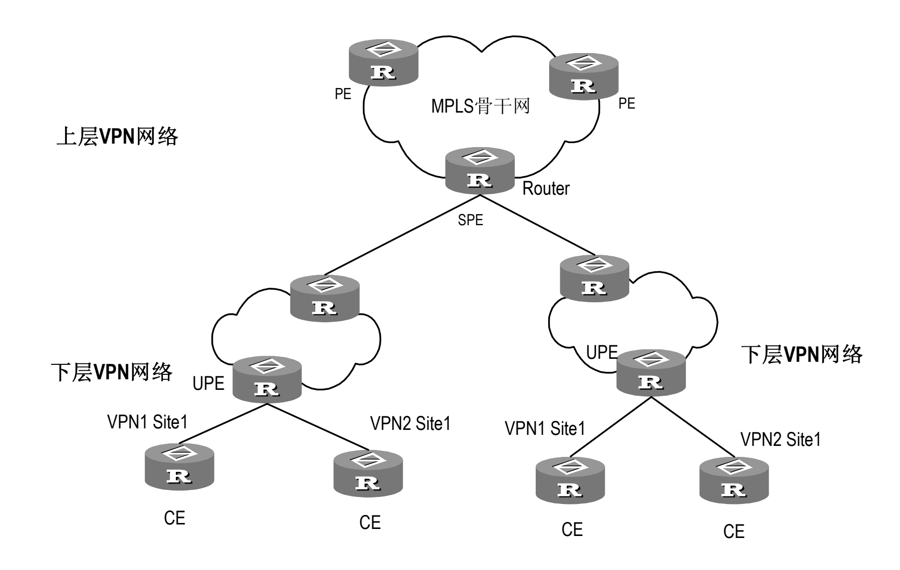


图3-4 分层式 BGP/MPLS VPN

3.1.4 多角色主机特性简介

从 CE 进入 PE 的报文的 VPN 属性由入接口绑定的 VPN 决定，这样，就实质上决定了由同一个入接口经过 PE 转发的所有 CE 设备必须都属于同一个 VPN。但是，在实际组网环境中，存在一个 CE 设备经过一个物理接口访问多个 VPN 的需求，这种需求也许可以通过设置不同的逻辑接口来实现，但是这种折中的解决方式会增加额外的配置负担，使用起来也有很大的局限性。为了解决该问题，利用多角色主机的构思，通过配置针对 IP 地址的策略路由来区分报文对不同 VPN 的访问，而从 PE 到 CE 的下行数据流，是通过静态路由来实现的，多角色主机情形下的静态路由跟普通的不一样，是通过一个 VPN 里面的静态路由指定其他 VPN 中的接口作为出接口来实现的，从而达到在一个逻辑接口访问多个 VPN 的目的。

3.1.5 OSPF 多实例简介

OSPF 是目前应用最为广泛的 IGP 路由协议之一，因此许多 VPN 将会使用 OSPF 作为其内部路由协议。如果在 PE-CE 链路上也使用 OSPF 将会非常便利：CE 路由器只需要支持 OSPF 协议，不需要支持更多的协议；同时，网络管理员也只需要了解 OSPF 协议；另外，如果客户需要将传统的 OSPF 骨干区域转换为 BGP/MPLS VPN 服务，那么在 PE 和 CE 之间使用 OSPF 可以简化这种转换。

因此，IETF 提出了两个新的 OSPF VPN 扩展草案，用以提供一个完整的方案，来解决 BGP/MPLS VPN 应用中将 OSPF 作为 PE-CE 间的路由协议的问题。此时，PE 路由器必须支持同时运行多个 OSPF 实例，此时，每一个 OSPF 实例与一个

VPN-Instance 相对应，拥有自己独立的接口、路由表，并且使用 BGP/OSPF 交互通过 MPLS 网络传送 VPN 的路由信息。

同时，支持 OSPF 多实例后，在一个路由器上可以运行多个 OSPF 进程，不同的进程可以绑定不同的VPN-Instance。在实际应用中，可以针对每种业务建立一个OSPF实例，采用 OSPF 多实例实现不同业务传输的完全隔离，低成本地解决传统局域网的安全性问题，极大的满足客户的需求。而 OSPF 多实例通常是运行在 PE 路由器上的，对于这种在局域网中运行 OSPF 多实例的路由器，通常也称之为 Multi-VPN-Instance CE。目前，局域网不同业务的隔离一般是通过交换机的 VLAN 功能实现的。OSPF Multi-VPN-Instance CE 提供了在路由器上实现业务隔离的方案。

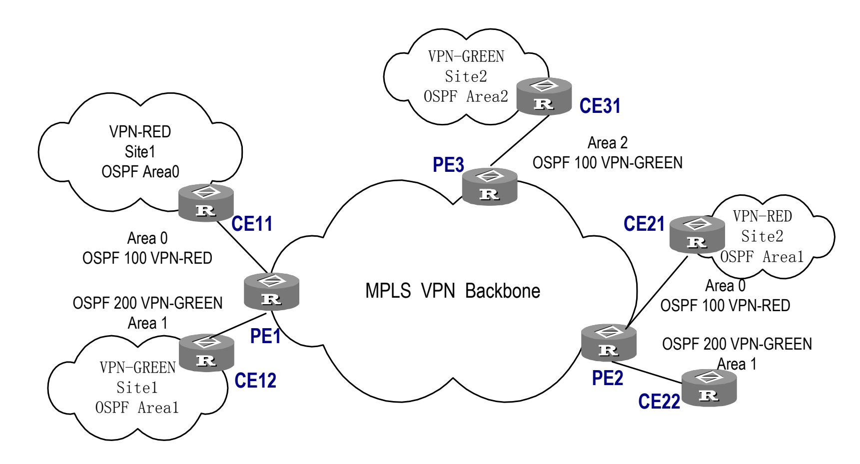


图3-5 OSPF 多实例在 MPLS/BGP VPN PE 中的应用

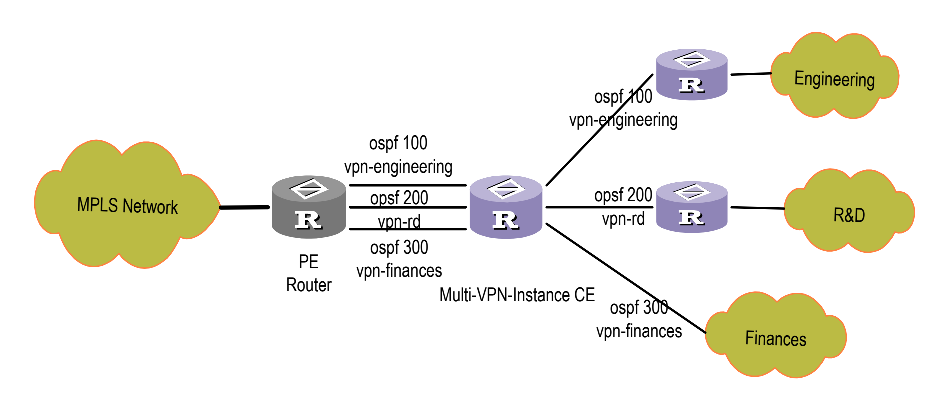


图3-6 Multi-VPN-Instance CE 在解决传统局域网安全性中的应用

3.2 BGP/MPLS VPN 配置

说明：

当同时配置了 L2VPN 业务和 L3VPN 业务时，以 L2VPN 业务为准；若此时去掉 L2VPN 业务，可以继续使用 L3VPN 业务。

要实现 BGP/MPLS VPN 的功能一般需要完成以下步骤：在 PE、CE、P 上配置基本信息；然后建立 PE 到 PE 的具有 IP 能力的逻辑或物理的链路；发布、更新 VPN 信息。

1. CE 设备

CE 设备的配置比较简单，只需配置静态路由、RIP、OSPF 或 EBGP 等，与相连的 PE 交换 VPN 路由信息。不需要配置 MPLS。

2. PE 设备

PE 设备的配置比较复杂，完成 MPLS/BGP VPN 的核心功能，大致可分为以下几个部分：

- 配置 MPLS 基本能力，与 P 设备和其它 PE 设备共同维护 LSP；
- 配置 BGP/MPLS VPN Site，即有关 vpn-instance 的配置；
- 配置静态路由、RIP、OSPF 或 MP-EBGP，与 CE 交换 VPN 路由信息；
- 配置 IGP，实现 PE 内部的互通；
- 配置 MP-IBGP，在 PE 之间交换 VPN 路由信息。

3. P 设备

P 设备的配置比较简单，主要是配置 MPLS 基本能力，提供 MPLS 转发能力和对 LDP 的支持。

下面 CE 对、PE、P 的配置进行分析。

3.2.1 CE 路由器的配置

CE 路由器作为用户端设备，仅需要做一些基本的配置，使之能够实现与 PE 设备进行路由信息的交换。目前可选择的路由交换方式有：静态路由、RIP、OSPF、EBGP、VLAN 子接口等。

1. 在 CE 上配置静态路由

如选择静态路由作为 CE-PE 间的路由交换方式，则应在 CE 上配置一条指向 PE 端的私网静态路由。

请在系统视图下进行下列配置。

表3-1 定义/清除 vpn-instance 路由表的静态路由

操作	命令
创建一条指定的 vpn-instance 的静态路由	**ip route-static** *ip-address* { *mask* | *mask-length* } { *interface-name* | *gateway-address* } [**preference** *preference-value*] [**reject** | **blackhole**]
删除一条指定的 vpn-instance 的静态路由	**undo ip route-static** *ip-address* { *mask* | *mask-length* } [*interface-name* | *gateway-address*] [**preference** *preference-value*]

每个静态路由都有缺省的优先值 60。配置静态路由时，可以指定一条静态路由的优先级。

2. 在 CE 上配置 RIP

如选择 RIP 作为 CE-PE 间的路由交换方式，则应在 CE 上配置 RIP。有关 RIP 配置步骤的详细说明，请参阅本手册路由协议中的 RIP 配置部分。

3. 在 CE 上配置 OSPF

如选择 OSPF 作为 CE-PE 间的路由交换方式，则应在 CE 上配置 OSPF。OSPF 的一般配置方法请参见《VRP3.3 操作手册》的“路由协议”部分。

当需要在 CE 路由器上隔离不同 VPN 之间的业务时，应配置 OSPF 多实例。此时的 CE 称为 Multi-VPN-Instance CE。

如果在 CE 路由器上配置 OSPF 进程绑定 VPN 实例后，则必须配置如下命令。

表3-2 配置路由器变为 Multi-VPN-Instance CE

操作	命令
配置路由器变为 Multi-VPN-Instance CE	**vpn-instance-capability simple**
取消配置	**undo vpn-instance-capability**

4. 在 CE 上配置 EBGP

如选择 BGP 作为 CE-PE 间的路由交换方式，则应在 CE 上配置 EBGP 对等体，并应引入直连路由、静态路由及其他 IGP 路由，以便 BGP 能够将 VPN 路由全部发布给 PE。

3.2.2 PE 路由器的配置

1. MPLS 基本能力配置

MPLS 基本能力配置包括配置 MPLS 的 LSR ID，全局使能 MPLS 并在所使用的接口视图下使能 MPLS。

具体配置请参考 MPLS 基本配置部分。

2. 定义 BGP/MPLS VPN Site

(1) 创建并进入 VPN 实例视图

vpn-instance 在实现中与 Site 关联，一个 Site 的 VPN 成员关系和路由规则等均在 vpn-instance 的配置下体现。

该命令用来创建新的 vpn-instance 或进入已有的 VPN 实例视图。如果该 vpn-instance 已存在，则直接进入该 vpn-instance 的视图下，进行相应的配置。

请在系统视图下进行下列配置。

表3-3 创建并进入 VPN 实例视图

操作	命令
创建并进入 VPN 实例视图	**ip vpn-instance** *vpn-instance-name*
删除 vpn-instance	**undo ip vpn-instance** *vpn-instance-name*

缺省情况下，没有定义 vpn-instance。

(2) 配置 VPN-instance 的 RD

在 PE 路由器上配置 RD，当从 CE 学习到的一条 VPN 路由引入 BGP 时，MP-BGP 将 RD 附加到 IPv4 前面，使之转换为 VPN IPv4 地址，使原来在 VPN 中的全局不唯一的 IPv4 地址成为全局唯一的 VPN IPv4 地址，以便在 VPN 中实现正确的路由。

请在 VPN 实例视图下进行下列配置。

表3-4 配置 VPN-instance 的 RD

操作	命令
配置 VPN-instance 的 RD	**route-distinguisher** *route-distinguisher*

参数无缺省值，一个 vpn-instance 只有配置了 RD 才能起作用。在配置 RD 之前，不能配置 vpn-instance 的其他任何参数。

(3) 为 vpn-instance 配置描述信息

为 VPN 实例配置描述信息，请在 VPN 实例视图下进行下列配置。

表3-5 为 vpn-instance 配置描述信息

操作	命令
为 vpn-instance 配置描述信息	**description** *vpn-instance-description*
删除 vpn-instance 描述信息	**undo description**

(4) 配置 vpn-instance 的 vpn-target 属性

VPN-target 用来控制 VPN 路由信息的发布，该属性是 BGP 的扩展团体属性。

VPN 路由的发布控制过程如下：

- 当从 CE 学习到的一条 VPN 路由引入 BGP 时，BGP 为它关联一个 VPN-Target 扩展团体属性列表，通常，这个列表是与 CE 相关联的 VPN-instance 的输出路由属性列表。
- VPN-instance 根据 VPN-target 中 **import-extcommunity** 定义输入路由属性列表，定义可被接受并引入此 VPN-instance 的路由范围。
- VPN-instance 根据 VPN-target 中的 **export-extcommunity** 对向外发布的路由进行 VPN-Target 属性的修改。

像 RD 一样，一个扩展团体或者是由一个自治系统号和一个任意的数组成，或者是由一个 IP 地址和一个任意的数组成。有以下两种格式：

与自治系统号（ASN）相关，形式为 16 位自治系统号(ASN)：32 位用户自定义数，例如：100:1。

与 IP 地址相关，形式为 32 位 IP 地址：16 位用户自定义数，例如：172.1.1.1:1。

请在 VPN 实例视图下进行下列配置。

表3-6 为 vpn-instance 创建 vpn-target 扩展团体

操作	命令
为 vpn-instance 创建 vpn-target 扩展团体	**vpn-target** *vpn-target-extcommunity* [**import-extcommunity** | **export-extcommunity** | **both**]
从 vpn-instance 关联的 vpn-target 列表中删除指定的 vpn-target 属性	**undo vpn-target** *vpn-target-extcommunity* [**import-extcommunity** | **export-extcommunity** | **both**]

系统缺省值为 **both**。一情况下 VPN 中的所有 Site 都可以互通，此时 import-extcommunity 和 export-extcommunity 属性相同，配 both 即可。

一条命令最多只能配置 16 个 vpn-target；一个 vpn-instance 最多只能配置 256 个 vpn-target。

(5) 限制一个 vpn-instance 中最大路由数

该命令可以限制一个 vpn-instance 中最大路由数，防止从一个 Site 引入的路由数量过多。

请在 VPN 实例视图下进行下列配置。

表3-7 限制一个 vpn-instance 中最大路由数

操作	命令
限制一个 vpn-instance 中最大路由数	**routing-table limit** { *warn threshold* | **simply-alert** }
取消最大路由数限制	**undo routing-table limit**

📖 说明：

改变 VPN-instance 的路由数目限制，不会影响已存在的路由表，如果需要新配置的路由限制立即生效，应重建相应的路由协议，或对相关接口进行 **shutdown/undo shutdown** 的操作。

(6) 将接口与 vpn-instance 关联

vpn-instance 通过与接口绑定实现与直接连接的 Site 相关联。当 SIte 发来的报文经此接口进入 PE 路由器，即可查找相应的 vpn-instance 获得路由信息（包括下一条、标签、输出接口等信息）。

该命令将一个 vpn-instance 与一个接口关联起来。

请在接口视图进行下列配置。

表3-8 将接口与 vpn-instance 关联

操作	命令
将接口与 vpn-instance 关联	**ip binding vpn-instance** *vpn-instance-name*
取消接口与 vpn-instance 关联	**undo ip binding vpn-instance** *vpn-instance-name*

 注意：

在实施接口与 vpn-instance 的绑定时，需要在接口上先执行 **ip binding vpn-instance** 命令**,**再配置 IP 地址，否则，已经配置的 IP 地址会因为执行 **ip binding vpn-instance** 命令而被删除。

3. 配置 PE 与 CE 间进行路由交换

目前 PE 与 CE 间的路由交换方式有：静态路由、RIP、OSPF、EBGP、VLAN 子接口等。

(1) 在 PE 上配置静态路由

可以在 PE 上配置一条指向 CE 端的静态路由，使 PE 通过静态路由的方式向 CE 学习 VPN 路由。

请在系统视图下进行下列配置。

表3-9 定义/清除 vpn-instance 路由表的静态路由

操作	命令
创建一条指定的 vpn-instance 的静态路由	**ip route-static vpn-instance** *vpn-instance-name1 vpn-instance-name2 … ip-address* { *mask* | *mask-length* } { *interface-name* | [**vpn-instance** *vpn-nexthop-name vpn-nexthop- address*] } [**preference** *preference-value*] [**reject** | **blackhole**]
删除一条指定的 vpn-instance 的静态路由	**undo ip route-static vpn-instance** *vpn-instance-name1 vpn-instance-name2 … ip-address* { *mask* | *mask-length* } { *interface-name* [**vpn-instance** *vpn-nexthop-name vpn-nexthop-address*] } [**preference** *preference-value*] [**reject** | **blackhole**]

每个静态路由都有缺省的优先值 60。配置静态路由时，可以指定一条静态路由的优先级。

(2) 在 PE 上配置 RIP 多实例

在 PE 和 CE 之间配置 RIP 时，需要在 PE 上指定 RIP 实例的运行环境。使用该命令进入路由实例的配置视图，并在此视图下配置 RIP 路由实例的引入、发布等。

请在 RIP 视图下进行下列配置。

表3-10 配置 PE、CE 间的 RIP 路由协议实例

操作	命令
创建 PE、CE 间的 RIP 路由协议实例	**ipv4-family** [**unicast**] **vpn-instance** *vpn-instance-name*
取消 PE、CE 间的 RIP 路由协议实例	**undo ipv4-family** [**unicast**] **vpn-instance** *vpn-instance-name*

然后配置 RIP 多实例引入 IBGP 路由。

有关 RIP 配置步骤的详细说明，请参阅本手册路由协议中的 RIP 配置部分。

(3) 在 PE 上配置 OSPF 多实例

在 PE 上运行 OSPF 与 CE 交换路由信息，需要 OSPF 支持多实例配置。其他配置，如 MPLS 基本配置、vpn-instance 的配置不变。需要注意的是，在 vpn-instance 下应引入 OSPF 路由及直连路由，在 OSPF 下也应引入 BGP 路由。这里对 OSPF 多实例的配置进行详细描述。

第一步：配置 OSPF 进程。

请在系统视图下进行下列配置。

表3-11 配置 OSPF 进程

操作	命令
配置一个 OSPF 进程	**ospf** *process-id* [**router-id** *router-id-number*] [**vpn-instance** *vpn-instance-name*]
删除一个 OSPF 进程	**undo ospf** *process-id*

缺省进程号为 1。

 注意：

一个 OSPF 进程只能属于一个 vpn 实例，一个实例可以包含多个 OSPF 进程。缺省状态下 OSPF 进程属于公网。

第二步：配置 domain-id。

Domain ID 用来标识不同的 OSPF 路由自治域，同一个 OSPF 路由域需要配置相同的 Domain ID。每个进程只能配置一个 domain-id，不同的进程之间可配置相同的 domain-id，也可配置不同的 domain-id。

表3-12 配置 domain-id

操作	命令
配置 domain-id	**domain-id** { *id-number* ｜ *id-addr* }
恢复缺省值	**undo domain-id**

id-number 缺省为 0，*id-addr* 缺省为 0.0.0.0。

一般建议：与每个 VPN 实例相关的网络中的所有 OSPF 实例要么配置一个相同的 **Domain ID**，要么都使用缺省的 0 作为 **Domain ID**。

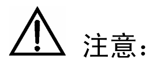
 注意：

执行这个命令后，不会马上起作用，只有在 reset ospf 后才会起作用。

第三步：配置标识 VPN 引入路由的 tag 值（可选）。

如果同一个 VPN site 链接了多个 PE，那么从 MPLS/BGP 学到的路由，被一个 PE 路由器通过 5 类或 7 类 LSA 向 VPN site 发布时，可能被另一个 PE 路由器收到，造成路由环。为了防止路由环，在这种情况下应配置 Route-tag，且同一 VPN 域的 PE 最好配置相同的 route-tag。

请在 OSPF 视图下进行下列配置。

 注意：

配置 route-tag 后，不会马上起作用，只有在 reset ospf 后才会起作用。

表3-13 配置标识 VPN 引入路由的 tag 值

操作	命令
配置标识 VPN 引入路由的 tag 值	**route-tag** *tag-number*
恢复缺省值	**undo route-tag**

tag-number 用来标识 VPN 引入路由的 tag 值，取默认值时前面两个字节为固定的 0xD000，后面的两个字节为本端 BGP 的 AS 号，比如本端 BGP AS 号为 100，则其默认的 tag 十进制值为 3489661028。该值的取值范围为 0～4294967295 的整数。

第四步：配置 sham link（可选）。

当两个 PE 之间存在 backdoor 链路（即不是通过 MPLS 骨干网络的 OSPF 链路），此时，又希望数据通过 MPLS 骨干传送时，就需要在两个 PE 之间配置 sham link。在 VPN PE 间配置 sham link，将其视为 OSPF 区域内的一条 link。其中，源地址和目的地址均为 32 位掩码的 loopback 接口地址，且该 loopback 接口需在 VPN 实例中绑定，并通过 BGP 引入直连路由引入到 BGP 中。

表3-14 配置 sham link

操作	命令
配置 sham link	**sham-link** *source-addr destination-addr* [**cost** *cost-value*] [**simple** *password* | **md5** *keyid key*] [**dead** *seconds*] [**hello** *seconds*] [**retransimit** *seconds*] [**trans-delay** *seconds*]
删除 sham link	**undo sham-link** *source-addr destination-addr*

其中，**cost** 的缺省值为 1，**dead** 的缺省值为 40 秒，**hello** 的缺省值为 10 秒，**retransmit** 的缺省值为 5 秒，**trans-delay** 的缺省值为 1 秒。

(4) 在 PE 上配置 EBGP

PE 与 CE 之间运行 EBGP，应在 MP-BGP 的 VPN-instance 视图下，为每个 VPN 配置邻居，并引入 CE 的 IGP 路由。

(5) 第一步：配置对等体组

请在配置 VPN-instance 视图下配置对等体所属的对等体组。

表3-15 配置对等体组

操作	命令
创建一个对等体组	**group** *group-name* [**internal** | **external]**
删除指定的对等体组	**undo group** *group-name*

配置时，如果不指定对等体组的配置类型，缺省为 internal。因为当 PE、CE 选用 BGP 进行路由交换时，一般属于不同的 AS，此时，应选用 **external**，建立 EBGP 对等体。

(6) 第二步：指定邻居 AS 号并向对等体组中加入组员

在PE和CE之间可以通过EBGP来交换路由信息，应为每个对应CE的vpn-instance 分别指定邻居的 AS 号。

请在 MP-BGP 的 vpn-instance 视图下进行下列配置。

表3-16 配置指定同伴的 AS 号

操作	命令
配置指定邻居的 AS 号	**peer** { *group-name* | [*peer-address* **group** *group-name*] } **as-number** *as-number*
删除邻居的 AS 号	**undo peer** { *group-name* | [*peer-address* **group** *grop-name*] } **as-number** *as-number*]

第三步：配置 MP-BGP 引入 IGP 路由

为了正确地在公网上传播 VPN 路由信息，PE 必须将与它直接相连的 CE 的 VPN 路由引入自己的 MBGP 路由表中，以发布给建立了 BGP 邻居关系的其它 PE。

举例来讲：如果 PE 与 CE 间使用静态路由，则 PE 需要在 MBGP 的 VPN-instance 视图下引入静态路由（import-route static）；如果 PE 与 CE 间运行 RIP，则 PE 需要在 MBGP 的 VPN-instance 视图下引入 RIP 路由（import-route rip），如果 PE 与 CE 之间运行 BGP，则 MBGP 直接引入直连路由即可。

请在 MBGP 的 VPN-instance 视图下进行下列配置。

表3-17 引入 IGP 协议的路由信息

操作	命令
配置 BGP 引入 IGP 协议的路由	**import-route** *protocol* [*process-id*] [**med** *med*]
取消 BGP 引入 IGP 协议的路由	**undo import-route** *protocol*

第四步：配置 BGP 的同步方式为不同步

请在 MBGP 的 VPN-instance 视图下进行下列配置。

表3-18 配置允许 BGP 与 IGP 不同步

操作	命令
配置允许 BGP 与 IGP 不同步	**undo synchronization**

目前，不同步方式为缺省配置。

第五步：Hub&Spoke 组网中允许路由环路配置（选配）。

一般来说，配置 PE-CE 的步骤通过指定邻居 AS 号后就可以完成，其余的配置保留系统默认值即可。

在标准 BGP 情况下，BGP 是通过 AS 号来检测路由循环，避免产生路由环路。但在 Hub&Spoke 组网方式下，如果在 PE 和 CE 之间运行 EBGP 协议，当 PE 将路由信息通告给 CE 时带上本自治系统的 AS 号。然后再从 CE 接收路由更新时，路由更新消息中就会带有本自治域系统的 AS 号，这样 PE 就不会接收这条路由更新信息。

通过配置 **peer allow-as-loop** 可以使 PE 路由器仍然接收从 CE 发送的包含本 AS 号的路由更新信息。

请在 IPv4 地址族子视图下进行下列配置。

表3-19 配置允许/禁止路由环路

操作	命令
配置允许路由环路	**peer** { *group-name* | *peer-address* } **allow-as-loop** *asn_limit*
配置禁止路由环路	**undo peer** { *group-name* | *peer-address* }**allow-as-loop** *asn_limit*

默认情况下，对于接收到的路由更新信息不允许产生环路信息。

第六步：配置 BGP 相关特性。

4. 配置 PE-PE 间进行路由交换

在 PE 上配置 MP-IBGP 协议，使得 PE 之间能够交互 VPN-IPv4 路由。

下列各项配置可在 BGP 视图或 VPN 实例视图下进行。

(1) 配置 IBGP

一般情况下，需要配置以下各项：

第一步：配置 BGP 的同步方式为不同步

第二步：配置 BGP 邻居

注意，应用 loopback 接口建立 BGP 的邻居关系，且子网掩码应为 32 位。

第三步：配置允许内部 BGP 会话使用任何可操作的 TCP 连接接口

一般情况下，BGP 是使用最佳本地地址进行 TCP 连接的；但为使接口在出现问题时该 TCP 连接还有效，可配置允许内部 BGP 会话使用任何可与对端建立 TCP 连接的接口，这条命令通常与 Loopback 接口一起配置。

表3-20 配置允许内部 BGP 会话使用任何可操作的 TCP 连接接口

操作	命令
配置允许内部 BGP 会话使用任何可操作的 TCP 连接的接口	**peer** { *peer-address* | *group-name* } **connect-interface** *interface-type interface-number*
恢复使用最佳本地地址进行 TCP 连接	**undo peer** { *peer-address* | *group-name* } **connect-interface** *interface-type interface-number*

BGP 可以使用指定接口和对端建立 BGP 邻居，一般来说都是指定 LoopBack 环回接口。因为这个接口永远处于 up 状态，可以减少由于网络震荡带来的冲击。

(2) 配置 MP-IBGP

第一步：进入协议地址族视图

请在 BGP 视图下进行下列配置。

表3-21 配置 vpnv 4 地址族

操作	命令
进入 MBGP 的 vpnv4 视图	**ipv4-family vpnv4** [**unicast**]
删除 MBGP 的 VPNv4 地址族视图的配置	**undo ipv4-family vpnv4** [**unicast**]

第二步：配置 MBGP 邻居。

在 MBGP 的 vpnv4 视图下配置 MBGP 的 **internal** 邻居。

第三步：配置激活对等体（组）。

由于缺省情况下，BGP 邻居处于激活状态，MBGP 邻居处于非激活状态。用户应在 vpnv4 视图下激活 MBGP 邻居。

第四步：配置在发布路由时将自身地址作为下一跳（可选）

由于该配置命令缺省为没有配置，所以在配置 PE-PE 之间的 MBGP 时，必须指明需要增加该配置命令。

表3-22 配置在发布路由时将自身地址作为下一跳

操作	命令
配置发布路由时将自身地址作为下一跳	**peer** { *peer-address* | *group-name* } **next-hop-local**
取消发布路由时将自身地址作为下一跳	**undo peer** { *peer-address* | *group-name* } **next-hop-local**

第五步：配置传送 BGP 更新报文时不携带私有自治系统号（可选）

表3-23 配置发送 BGP 更新报文时不携带私有自治系统号

操作	命令
配置发送 BGP 更新报文时不携带私有自治系统号	**peer** { *peer-address* | *group-name* } **public-as-only**
配置发送 BGP 更新报文时携带私有自治系统号	**undo peer** { *peer-address* | *group-name* } **public-as-only**

第六步：配置向对等体（组）发送缺省路由

该命令一般用于分层 PE 的 SPE 上，以增加一条下一跳为自身的缺省路由。

表3-24 配置向对等体（组）发送缺省路由

操作	命令
配置向对等体（组）发送缺省路由	**peer** { *peer-address* | *group-name* } **default-route-advertise** [**vpn-instance** *vpn-instancename*]
取消向对等体（组）发送缺省路由	**undo peer** { *peer-address* | *group-name* } **default-route-advertise** [**vpn-instance** *vpn-instancename*]

第七步：配置 BGP 邻居作为分层 BGP/MPLS VPN 的 UPE

此命令仅用于分层 BGP/MPLS VPN 的 UPE。

请在 ipv4 地址族视图(vpnv4)下配置下列命令。

表3-25 配置 BGP 邻居作为分层 BGP/MPLS VPN 的 UPE

操作	命令
配置 BGP 邻居作为分层 BGP/MPLS VPN 的 UPE	**peer** *peer-address* **upe**
取消该配置	**undo peer** *peer-address* **upe**

(3) 多角色主机特性配置

必须在 PE 上进行如下配置，才能够实现与之相连的 Site 的多角色应用。

第一步：在 PE 上配置 vpn-instance。

在多角色主机应用中，一个Site经一个物理接口接入到PE，但可以访问多个VPN，故需要在PE上配置多个vpn-instance，仅将可以直接访问的VPN与连接Site的接口绑定即可。vpn-instance的配置，请参考 2. 定义BGP/MPLS VPN Site。

第二步：配置策略路由。

在策略路由使能后满足 route-policy 的条件情况下，指定报文依次在已经配置的 *vpn-name1*，*vpn-name2*，*vpn-name3*，*vpn-name4*，*vpn-name5*，*vpn-name6* 中查找私网转发路由并进行相应的转发。

请在 route-policy 视图下进行下列配置。

表3-26 在指定 VPN 实例中查找私网转发路由并进行相应的转发

操作	命令
在指定 VPN 实例中查找私网转发路由并进行相应的转发	**apply access-vpn vpn-instance** { *vpn-name1 vpn-name2* … }
取消在指定 VPN 实例中的查找	**undo apply access-vpn vpn-instance** { *vpn-name1 vpn-name2* … }

第三步：配置到一个私网的静态路由。

通过配置静态路由，指定其他 VPN 中的接口作为出接口，从而使从 PE 返回到 CE 的下行报文，能够直接返回 Site。

请在系统视图下配置下列命令。

表3-27 配置到一个私网的静态路由

操作	命令
用于配置到一个私网的静态路由	**ip route-static vpn-instance** *vpn-instance-name1 vpn-instance-name2* … *ip-address* { *mask* | *mask-length* } { *interface-name* | [**vpn-instance** *vpn-nexthop-name vpn-nexthop-address*] } [**preference** *preference-value*] [**reject** | **blackhole**]
取消静态路由的配置	**undo ip route-static vpn-instance** *vpn-instance-name1 vpn-instance-name2* … *ip-address* { *mask* | *mask-length* } { *interface-name* [**vpn-instance** *vpn-nexthop-name vpn-nexthop-address*] } [**preference** *preference-value*] [**reject** | **blackhole**]

3.2.3 配置 P 路由器

P 路由器不需要维护 VPN 路由，但需要保证公网的连通性，并配合 PE 建立 LSP，故需要如下配置：

第一步：配置MPLS基本能力，并在P与PE相连的各接口上使能LDP，以转发MPLS报文。具体配置请参见 第 2 章 MPLS基本能力配置。

第二步：在 P 与 PE 相连的各接口上启动 OSPF 协议，并引入直连路由，实现 PE 内部的互通。具体配置请参见“路由协议”的“OSPF“部分。

3.3 BGP/MPLS VPN 显示与调试

1. 从 BGP 表中显示 VPN 地址信息

在完成上述配置后，在所有视图下执行 **display** 命令，可以显示 BGP 数据库中 VPNv4 信息，通过查看显示信息，验证配置的效果。

表3-28 从 BGP 表中显示 VPN 地址信息

操作	命令
从 BGP 表中显示 VPN 地址信息	**display bgp vpnv4** { **all** | **route-distinguisher** *rd-value* | **vpn-instance** *vpn-instance-name* } { **group** | **network** | **peer** | **routing-table** }

2. 显示与 vpn-instance 相关联的 IP 路由表

在完成上述配置后，在所有视图下执行 **display** 命令可以显示与 vpn-instance 相关联的 IP 路由表中的相关信息，通过查阅该显示信息来验证配置的效果。

表3-29 显示与 vpn-instance 相关联的 IP 路由表

操作	命令
显示与 vpn-instance 相关联的 IP 路由表	**display ip routing-table vpn-instance** *vpn-instance-name* [*ip-address*] [**verbose**]

3. 显示 vpn-instance 相关信息

在完成上述配置后，在所有视图下执行 **display** 命令可以显示 vpn-instance 相关信息，包括该 VPN 实例的 RD，描述以及与之关联的接口等信息，通过查阅该显示信息来验证配置的效果。

表3-30 显示 vpn-instance 相关信息

操作	命令
显示 vpn-instance 相关信息，包括该 VPN 实例的 RD，描述以及与之关联的接口等信息	**display ip vpn-instance** [*vpn-instance-name* | **verbose**]

4. 显示涉及处理 BGP 的调试信息

在用户视图下，执行 **debugging** 命令可对 BGP 相关运行情况进行调试。

表3-31 显示涉及处理 BGP 的调试信息

操作	命令
显示涉及处理 BGP 的调试信息	**debugging bgp** { { **keepalive** | **open** | **packet** | **update** | **route-refresh** } [**receive** | **send** | **verbose**] } { **all** | **event** | **normal** }
关闭调试信息	**undo debugging bgp** { { **keepalive** | **open** | **packet** | **update** | **route-refresh** } [**receive** | **send** | **verbose**] } { **all** | **event** | **normal** }

5. 显示 MPLS l3vpn-lsp 的相关信息

表3-32 显示 MPLS l3vpn-lsp 的相关信息

操作	命令
显示 mpls l3vpn 标签交换路径的相关信息。	**display mpls l3vpn-lsp** [**verbose**] **include** *text*
显示 mpls l3vpn 标签交换路径 vpn-instance 相关的信息。	**display mpls l3vpn-lsp** [**vpn-instance** *vpn-instance-name*] [**transit | egress | ingress**] [**include** *text* | **verbose**]

6. 显示配置的 sham link

表3-33 显示配置的 sham link

操作	命令
显示配置的 sham link	**display ospf** [*process-id*] **sham-link**

3.4 BGP/MPLS VPN 典型配置举例

3.4.1 BGP/MPLS VPN 综合组网举例

1. 组网需求

- CE1、CE3 构成 VPN-A，CE2、CE4 构成 VPN-B；
- 不同 VPN 用户之间不能互相访问。VPN-A 使用的 VPN-target 属性为 111:1，VPN-B 使用的 VPN-target 属性为 222:2；
- PE 使用 Quidway 路由器，P 为支持 MPLS 的 Quidway 路由器，CE 为一般的中低端路由器。

📖 说明：

此案例的配置重点在于：

- CE 与 PE 之间配置 EBGP 交换 VPN 路由信息
- PE 之间配置 OSPF 实现 PE 内部的互通
- PE 之间配置 MP-IBGP 交换 VPN 路由信息

2. 组网图

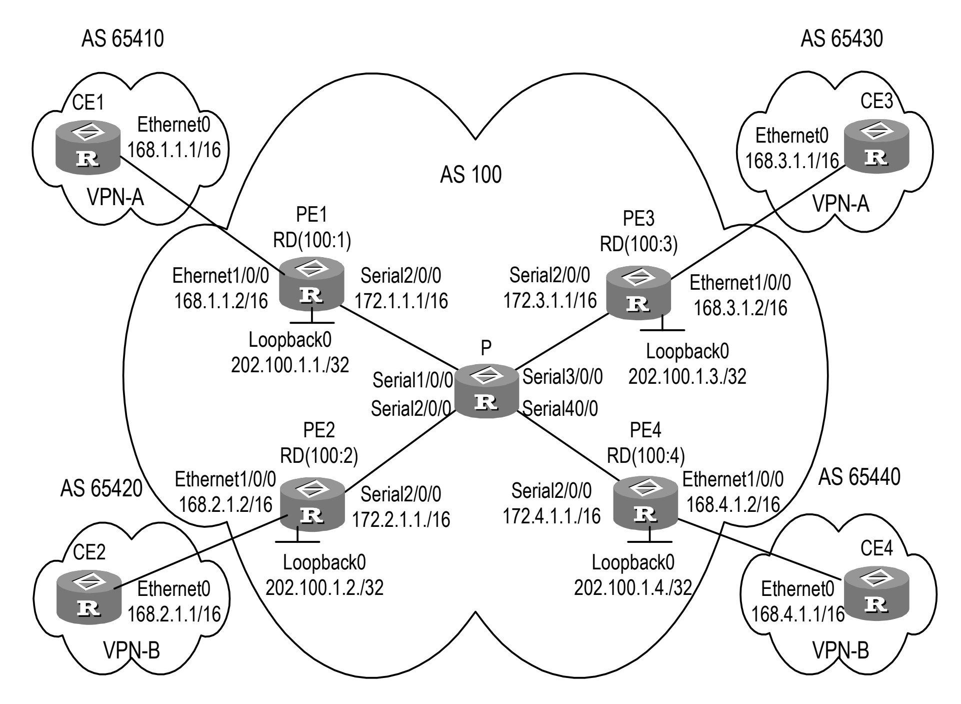


图3-7 BGP/MPLS VPN 综合组网图

3. 配置步骤

下面依次介绍各 PE 路由器、CE 路由器和 P 路由器上的配置。

(1) 配置 CE1

CE1 与 PE1 建立 EBGP 邻居，引入直连路由和静态路由，从而将 CE1 的内部 VPN 路由引入到 BGP，进而发布给 PE1。CE1 使用接口 Ethernet0 与 PE1 相连。

```
[CE1] interface ethernet 0/0/0
[CE1-Ethernet0/0/0] ip address 168.1.1.1 255.255.0.0
[CE1-Ethernet0/0/0] quit
[CE1] bgp 65410
[CE1-bgp] group 168 external
[CE1-bgp] peer 168.1.1.2 group 168 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] import-route static
```

📖 **说明：**

另外 3 个 CE 路由器（CE2～CE4）配置与 CE1 路由器配置类似，配置过程省略。

(2) 配置 PE1

在 PE1 上创建 VPN-A 的 VPN-instance，并配置相关属性以控制 VPN 路由信息的发布。

```
[PE1] ip vpn-instance vpna
[PE1-vpn-vpna] route-distinguisher 100:1
[PE1-vpn- vpna] vpn-target 111:1 both
[PE1-vpn- vpna] quit
```

PE1 与 CE1 间建立 EBGP 邻居，并将学到的 CE1 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-af-vpn-instance] group 168 external
[PE1-bgp-af-vpn-instance] peer 168.1.1.1 group 168 as-number 65410
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] quit
```

将 PE1 与 CE1 相连的接口 Ethernet1/0/0 绑定到 VPN-A（需要注意的是，应先配置接口与 VPN-instance 的关联后，再配置接口的 IP 地址）。

```
[PE1] interface ethernet 1/0/0
[PE1-Ethernet1/0/0] ip binding vpn-instance vpna
[PE1-Ethernet1/0/0] ip address 168.1.1.2 255.255.0.0
[PE1-Ethernet1/0/0] quit
```

配置 LoopBack 接口（对 PE 路由器，配置 LoopBack 接口地址时，必须使用 32 位掩码的主机地址，以防止此路由被聚合，导致 LSP 不能正确处理内层标签）。

```
[PE1] interface loopback0
[PE1-LoopBack 0] ip address 202.100.1.1 255.255.255.255
[PE1-LoopBack 0] quit
```

配置 MPLS 基本能力，并在 PE1 与 P 路由器相连的接口上使能 MPLS 及 LDP。建立 LSP 和实现 MPLS 报文转发。

```
[PE1] mpls lsr-id 202.100.1.1
[PE1] mpls
[PE1] mpls ldp
[PE1] interface Serial2/0/0
[PE1-Serial2/0/0] ip address 172.1.1.1 255.255.0.0
[PE1-Serial2/0/0] mpls
```

```
[PE1-Serial2/0/0] mpls ldp enable
[PE1-Serial2/0/0] quit
```

在 PE1 与 P 路由器相连的接口及 loopback 接口上启用 OSPF，并引入直连路由。实现 PE 内部的互通。

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 172.1.0.0 0.0.255.255
[PE1-ospf-1-area-0.0.0.0] network 202.100.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] import-route direct
[PE1-ospf-1] quit
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE1] bgp 100
[PE1-bgp] group 202 internal
[PE1-bgp] peer 202.100.1.3 group 202
[PE1-bgp] peer 202.100.1.3 connect-interface loopback0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer 202 enable
[PE1-bgp-af-vpn] peer 202.100.1.3 group 202
[PE1-bgp-af-vpn] quit
[PE1-bgp] quit
```

(3) 配置 P

配置 P 路由器的 MPLS 基本能力，并在 P 与 PE 相连的各接口上使能 LDP，以转发 MPLS 报文。

```
[P] mpls lsr-id 172.1.1.2
[P] mpls
[P] mpls ldp
[P] interface Serial1/0/0
[P-Serial1/0/0] ip address 172.1.1.2 255.255.0.0
[P-Serial1/0/0] mpls
[P-Serial1/0/0] mpls ldp enable
[P-Serial1/0/0] interface Serial2/0/0
[P-Serial2/0/0] ip address 172.2.1.2 255.255.0.0
[P-Serial2/0/0] mpls
[P-Serial2/0/0] mpls ldp enable
[P-Serial2/0/0] interface Serial3/0/0
[P-Serial3/0/0] ip address 172.3.1.2 255.255.0.0
[P-Serial3/0/0] mpls
[P-Serial3/0/0] mpls ldp enable
```

```
[P-Serial3/0/0] interface Serial4/0/0
[P-Serial4/0/0] ip address 172.4.1.2 255.255.0.0
[P-Serial4/0/0] mpls
[P-Serial4/0/0] mpls ldp enable
[P-Serial4/0/0] quit
```

在 P 与 PE 相连的各接口上启动 OSPF 协议，并引入直连路由，实现 PE 内部的互通。

```
[P] ospf
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] network 172.1.1.0 0.0.255.255
[P-ospf-1-area-0.0.0.0] network 172.2.1.0 0.0.255.255
[P-ospf-1-area-0.0.0.0] network 172.3.1.0 0.0.255.255
[P-ospf-1-area-0.0.0.0] network 172.4.1.0 0.0.255.255
[P-ospf-1-area-0.0.0.0] quit
[P-ospf-1] import-route direct
```

(4) 配置 PE3

说明：

PE3 的配置过程与 PE1 相似，需要注意 PE3 上对于 VPN 路由属性的设置，以了解如何控制同一 VPN 的路由信息在 MPLS 网上的发布（相同的 VPN-target）。

在 PE3 上创建 VPN-A 的 VPN-instance，并配置相关属性以控制 VPN 路由信息的发布。

```
[PE3] ip vpn-instance vpna
[PE3-vpn-vpna] route-distinguisher 100:3
[PE3-vpn-vpna] vpn-target 111:1 both
[PE3-vpn-vpna] quit
```

PE3 与 CE3 间建立 EBGP 邻居，并将学到的 CE3 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpna
[PE3-bgp-af-vpn-instance] group 168 external
[PE3-bgp-af-vpn-instance] peer 168.1.1.1 group 168 as-number 65430
[PE3-bgp-af-vpn-instance] import-route direct
[PE3-bgp-af-vpn-instance] quit
[PE3-bgp] quit
```

将 PE3 与 CE3 相连的接口 Ethernet1/0/0 绑定到 VPN-A。

```
[PE3] interface ethernet 1/0/0
[PE3-Ethernet1/0/0] ip binding vpn-instance vpna
```

```
[PE3-Ethernet1/0/0] ip address 168.3.1.2 255.255.0.0
[PE3-Ethernet1/0/0] quit
```

配置 LoopBack 接口。

```
[PE3] interface loopback0
[PE3-LoopBack 0] ip address 202.100.1.3 255.255.255.255
[PE3-LoopBack 0] quit
```

配置 MPLS 基本能力，并在 PE3 与 P 路由器相连的接口上使能 MPLS 及 LDP。建立 LSP 和实现 MPLS 报文转发。

```
[PE3] mpls lsr-id 202.100.1.3
[PE3] mpls
[PE3] mpls ldp
[PE3] interface Serial 2/0/0
[PE3-Serial2/0/0] ip address 172.3.1.1 255.255.0.0
[PE3-Serial2/0/0] mpls
[PE3-Serial2/0/0] mpls ldp enable
[PE3-Serial2/0/0] quit
```

在 PE3 与 P 路由器相连的接口及 loopback 接口上启用 OSPF，并引入直连路由。

```
[PE3] ospf
[PE3-ospf-1] area 0
[PE3-ospf-1-area-0.0.0.0] network 172.3.0.0 0.0.255.255
[PE3-ospf-1-area-0.0.0.0] network 202.100.1.3 0.0.0.0
[PE3-ospf-1-area-0.0.0.0] import-route direct
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。

```
[PE1] bgp 100
[PE1-bgp] group 202 internal
[PE3-bgp] peer 202.100.1.1 group 202
[PE3-bgp] peer 202.100.1.1 connect-interface loopback0
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpn] peer 202 enable
[PE3-bgp-af-vpn] peer 202.100.1.1 group 202
[PE3-bgp-af-vpn] quit
```

(5) PE2 与 PE4 的配置

PE2 与 PE4 的配置与 PE1 和 PE3 的配置类似，具体配置过程略。

3.4.2 采用 GRE 隧道的 BGP/MPLS VPN 配置举例

1. 组网需求

- CE-1、CE-3 构成 VPN_A，CE-2、CE-4 构成 VPN_B。

- PE 使用的路由器为 Quidway 系列化路由器，PE 需要支持 MPLS 能力。P 为任意的 Quidway 系列化路由器（无须支持 MPLS 能力），CE 为一般的中低端路由器。

2. 组网图

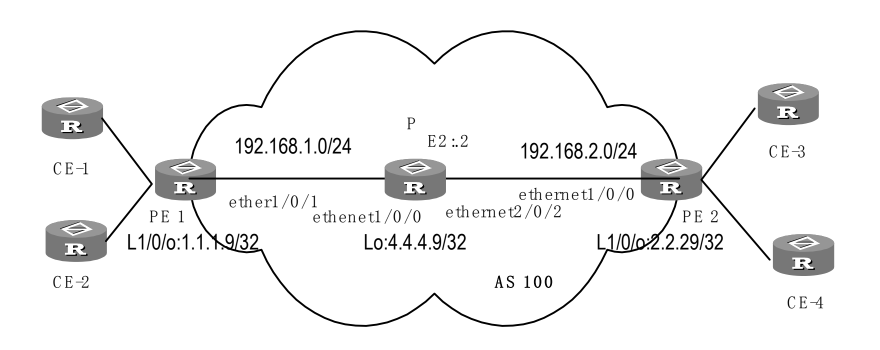


图3-8 采用 GRE 隧道的 BGP/MPLS VPN 组网图

3. 配置步骤

(1) 配置 CE1

1 与 PE1 建立 EBGP 邻居，引入直连路由和静态路由，从而将 CE1 的内部 VPN 路由引入到 BGP，进而发布给 PE1，CE1 使用接口 Ethernet0/0/0 与 PE1 相连。

```
[CE1] interface ethernet 0/0/0
[CE1-Ethernet0/0/0] ip address 20.1.1.1 255.255.0.0
[CE1-Ethernet0/0/0] quit
[CE1] bgp 65410
[CE1-bgp] group 20 external
[CE1-bgp] peer 20.1.1.2 group 20 as-number 100
[CE1-bgp] import-route direct
[CE1-bgp] import-route static
```

说明：

另外 3 个 CE 路由器（CE2~CE4）配置与 CE1 路由器配置类似，配置过程省略。

(2) 配置 PE1

VPN-instance 配置。

```
[PE1] ip vpn-instance vpna
[PE1-vpn-vpna] route-distinguisher 100:1
[PE1-vpn-vpna] vpn-target 100:1 both
[PE1-vpn-vpna] VPN-target 100:2 import-extcommunity
[PE1-vpn-vpna] VPN-target 100:3 export-extcommunity
```

```
[PE1-vpn-vpna] quit
```

接口配置，将 PE1 与 CE1 相连的接口 Ethernet1/0/0 绑定到 VPN-A。

```
[PE1] interface loopback0
[PE1-LoopBack0] ip address 1.1.1.9 255.255.255.255
[PE1-LoopBack0] quit
[PE1] interface ethernet 1/0/0
[PE1-Ethernet1/0/0] ip binding vpn-instance vpna
[PE1-Ethernet1/0/0] ip address 20.1.1.2 255.255.0.0
[PE1-LoopBack1/0/0] quit
[PE1] interface ethernet1/0/1
[PE1-Ethernet1/0/1] ip address 192.168.1.1 255.255.255.0
[PE1-Ethernet1/0/1] quit
```

PE1 与 CE1 间建立 EBGP 邻居，并引入 VPN-instance 的接口路由。

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-af-vpn-instance] group 20 external
[PE1-bgp-af-vpn-instance] peer 20.1.1.1 group 20 as-number 65410
[PE1-bgp-af-vpn-instance] peer 20 next-hop-local
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] quit
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换，并在 VPNv4 地址族视图下激活 IBGP 对等体。

```
[PE1] bgp 100
[PE1-bgp] group 2
[PE1-bgp] peer 2.2.2.9 group 2
[PE1-bgp] peer 2.2.2.9 connect-interface loopback0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer 2 enable
[PE1-bgp-af-vpn] peer 2.2.2.9 group 2
```

在 PE1 与 P 路由器相连的接口及环回接口上启用 OSPF，并引入直连路由，实现 PE 内部的互通。

```
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.255.255.255
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.9 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] import-route direct
```

配置 MPLS。

```
[PE1] mpls lsr-id 172.1.1.1
[PE1] mpls
```

```
[PE1] mpls ldp
```

配置 GRE 隧道。

```
[PE1] interface tunnel 1
[PE1-Tunnel1] tunnel-protocol gre
[PE1-Tunnel1] source loopback 0
[PE1-Tunnel1] destination 2.2.2.9
```

说明：

PE2 配置与 PE1 类似，配置过程省略。

---

(3) 配置 P

配置接口。

```
[P] interface ethernet1/0/0
[P-ethernet1/0/0] ip address 192.168.1.2 255.255.255.0
[P] interface ethernet2/0/1
[P-ethernet2/0/1] ip address 192.168.2.2 255.255.255.0
```

配置 OSPF 协议。

```
[P] ospf
[P-ospf-1] area 0
[P-ospf-1-area-0.0.0.0] network 192.168.1.0 0.255.255.255
[P-ospf-1-area-0.0.0.0] network 192.168.2.0 0.255.255.255
[P-ospf-1-area-0.0.0.0] import-route direct
```

3.4.3 Extranet 组网举例

1. 组网需求

公司 A 和公司 B 通过 VPN 互联，两个公司的总部都在城市 C，虚拟内部网号分别为 VPN1 和 VPN2。

通过 MPLS 给用户提供 VPN 功能，两个 VPN 之间有一部分共享资源在城市 C，两个 VPN 内的用户都可以访问位于城市 C 的资源，但城市 A 和 B 的 VPN 用户不能互相访问。

由于两个公司在 PE-C 上共用一个 VPN-instance，所以两个公司使用的 IP 地址空间不能重叠。

说明：

此案例的配置重点在于：通过对不同 PE 上 VPN-target 属性的设置，控制不同城市 VPN 用户对资源的访问权限。

---

2. 组网图

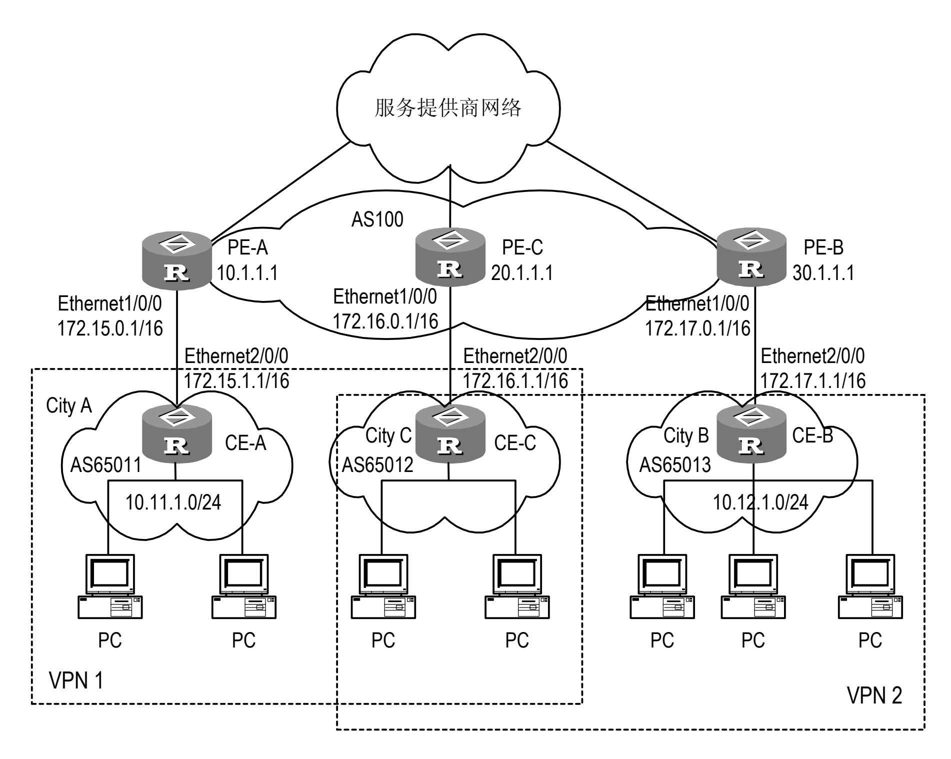


图3-9 Extranet 组网图

3. 配置步骤

说明：

本配置步骤部分省略了 PE 路由器之间的 MPLS 基本能力配置、PE 路由器与 P 路由器间的配置、以及 CE 路由器的配置。这部分的内容可以参考前一个配置举例。

---

(1) 配置 PE-A

在 PE-A 上创建 VPN1 的 VPN-instance，能够收发 VPN-target 为 111:1 的 VPN 路由信息。

```
[PE-A] ip vpn-instance 1
[PE-A-vpn-1] route-distinguisher 100:1
[PE-A-vpn-1] vpn-target 111:1 both
[PE-A-vpn-1] quit
```

PE-A 与 CE-A 间建立 EBGP 邻居，并将学到的 CE-A 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE-A] bgp 100
[PE-A-bgp] ipv4-family vpn-instance vpn-instance1
[PE-A-bgp-af-vpn-instance] group 172
[PE-A-bgp-af-vpn-instance] peer 172.15.1.1 group 172 as-number 65011
```

```
[PE-A-bgp-af-vpn-instance] import-route direct
[PE-A-bgp-af-vpn-instance] import-route static
[PE-A-bgp-af-vpn-instance] quit
[PE-A-bgp] quit
```

将与 CE-A 连接的接口 Ethernet1/0/0 与 VPN-instance1 绑定。

```
[PE-A] interface ethernet 1/0/0
[PE-A-Ethernet1/0/0] ip binding vpn-instance vpn-instance1
[PE-A-Ethernet1/0/0] ip address 172.15.0.1 255.255.0.0
[PE-A-Ethernet1/0/0] quit
```

配置 LoopBack 接口。

```
[PE-A] interface loopback 0
[PE-A-LoopBack0] ip address 10.1.1.1 255.255.255.255
[PE-A-LoopBack0] quit
```

配置 MPLS 基本能力。

```
[PE-A] mpls lsr-id 10.1.1.1
[PE-A] mpls
[PE-A] mpls ldp
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE-A] bgp 100
[PE-A-bgp] group 20
[PE-A-bgp] peer 20.1.1.1 group 20
[PE-A-bgp] peer 20.1.1.1 connect-interface loopback 0
[PE-A-bgp] group 30
[PE-A-bgp] peer 30.1.1.1 group 30
[PE-A-bgp] peer 30.1.1.1 connect-interface loopback 0
[PE-A-bgp] ipv4-family vpnv4
[PE-A-bgp-af-vpn] peer 20 enable
[PE-A-bgp-af-vpn] peer 20.1.1.1 group 20
[PE-A-bgp-af-vpn] peer 30 enable
[PE-A-bgp-af-vpn] peer 30.1.1.1 group 30
[PE-A-bgp-af-vpn] quit
```

(2) 配置 PE-C

在 PE-C 上创建一个 VPN-instance，能够收发 VPN-target 为 111:1 和 222:2 的 VPN 路由信息。

```
[PE-C] ip vpn-instance 2
[PE-C-vpn-2] route-distinguisher 100:2
[PE-C-vpn-2] vpn-target 111:1
[PE-C-vpn-2] vpn-target 222:2
```

```
[PE-C-vpn-2] quit
```

PE-C 与 CE-C 间建立 EBGP 邻居，并将学到的 CE-C 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE-C] bgp 100
[PE-C-bgp] ipv4-family vpn-instance vpn-instance2
[PE-C-bgp-af-vpn-instance] group 172
[PE-C-bgp-af-vpn-instance] peer 172.16.1.1 group 172 as-number 65012
[PE-C-bgp-af-vpn-instance] import-route direct
[PE-C-bgp-af-vpn-instance] import-route static
[PE-C-bgp-af-vpn-instance] quit
[PE-C-bgp] quit
```

将与 CE-C 相连的接口 Ethernet1/0/0 与 VPN-instance2 绑定。

```
[PE-C] interface ethernet 1/0/0
[PE-C-Ethernet1/0/0] ip binding vpn-instance vpn-instance2
[PE-C-Ethernet1/0/0] ip address 172.16.0.1 255.255.0.0
```

配置 LoopBack 接口。

```
[PE-C] interface loopback 0
[PE-C-LoopBack0] ip address 30.1.1.1 255.255.255.255
[PE-C-LoopBack0] quit
```

配置 MPLS 基本能力。

```
[PE-C] mpls lsr-id 20.1.1.1
[PE-C] mpls
[PE-C] mpls ldp
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE-C] bgp 100
[PE-C-bgp] group 10
[PE-C-bgp] peer 10.1.1.1 group 10
[PE-C-bgp] peer 10.1.1.1 connect-interface loopback 0
[PE-C-bgp] group 30
[PE-C-bgp] peer 30.1.1.1 group 30
[PE-C-bgp] peer 30.1.1.1 connect-interface loopback 0
[PE-C-bgp] ipv4-family vpnv4
[PE-C-bgp-af-vpn] peer 10 enable
[PE-C-bgp-af-vpn] peer 10.1.1.1 group 10
[PE-C-bgp-af-vpn] peer 30 enable
[PE-C-bgp-af-vpn] peer 30.1.1.1 group 30
[PE-C-bgp-af-vpn] quit
```

(3) 配置 PE-B

在 PE-B 上创建 VPN2 的 VPN-instance，能够收发 VPN-target 为 222:2 的 VPN 路由信息。

```
[PE-B] ip vpn-instance 3
[PE-B-vpn-3] route-distinguisher 100:3
[PE-B-vpn-3] vpn-target 222:2
[PE-B-vpn-3] quit
```

PE-B 与 CE-B 间建立 EBGP 邻居，并将学到的 CE-B 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE-B] bgp 100
[PE-B-bgp] ipv4-family vpn-instance vpn-instance3
[PE-B-bgp-af-vpn-instance] group 172 external
[PE-B-bgp-af-vpn-instance] peer 172.17.1.1 group 172 as-number 65013
[PE-B-bgp-af-vpn-instance] import-route direct
[PE-B-bgp-af-vpn-instance] import-route static
[PE-B-bgp-af-vpn-instance] quit
[PE-B-bgp] quit
```

将与 CE-B 相连的接口 Ethernet1/0/0 与 VPN-instance3 绑定。

```
[PE-B] interface ethernet 1/0/0
[PE-B-Ethernet1/0/0] ip binding vpn-instance vpn-instance3
[PE-B-Ethernet1/0/0] ip address 172.17.0.1 255.255.0.0
[PE-B-Ethernet1/0/0] quit
```

配置 LoopBack 接口。

```
[PE-B] interface loopback 0
[PE-B-LoopBack0] ip address 20.1.1.1 255.255.255.255
[PE-B-LoopBack0] quit
```

配置 MPLS 基本能力。

```
[PE-B] mpls lsr-id 30.1.1.1
[PE-B] mpls
[PE-B] mpls ldp
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE-B] bgp 100
[PE-B-bgp] group 10
[PE-B-bgp] peer 10.1.1.1 group 10
[PE-B-bgp] peer 10.1.1.1 connect-interface loopback 0
[PE-B-bgp] group 20
[PE-B-bgp] peer 20.1.1.1 group 20
[PE-B-bgp] peer 20.1.1.1 connect-interface loopback 0
[PE-B-bgp] ipv4-family vpnv4
```

```
[PE-B-bgp-af-vpn] peer 10 enable
[PE-B-bgp-af-vpn] peer 10.1.1.1 group 10
[PE-B-bgp-af-vpn] peer 20 enable
[PE-B-bgp-af-vpn] peer 20.1.1.1 group 20
[PE-B-bgp-af-vpn] quit
```

3.4.4 Hub&Spoke 组网举例

1. 组网需求

Hub&Spoke 组网方式也称为中心服务器拓扑组网。中心 Site 称为 Hub-Site，它知道同一 VPN 所有其它 Site 的路由；不处于中心的 site 称为 Spoke-Site，它们的流量通过 HUB-Site 到达目的地。Hub-Site 是 Spoke-Site 的中枢节点。

某银行的网络包括各分公司网络与公司总部的网络，要求各分公司之间不能直接交换数据，必须通过总部进行通信，以进行统一控制。采用 Hub&Spoke 拓扑，CE2 和 CE3 为 Spoke 站点，CE1 为银行数据中心 Hub 站点，CE2 与 CE3 间的通信由 CE1 控制。

- PE1 分别与 PE2、PE3 建立 IBGP 邻居关系，但 PE2 与 PE3 不建立 IBGP 邻居关系，不交换 VPN 路由信息；
- 在 PE1 上创建两个 VPN-instance，引入 VPN-target 属性为 100:1 的 VPN 路由，对发布的 VPN 路由设置 VPN-target 属性 100:2；
- 在 PE2 上创建一个 VPN-instance，引入 VPN-target 属性为 100:2 的 VPN 路由，对发布的 VPN 路由设置 VPN-target 属性 100:1；
- 在 PE3 上创建一个 VPN-instance，引入 VPN-target 属性为 100:2 的 VPN 路由，对发布的 VPN 路由设置 VPN-target 属性 100:1。

经过以上配置，PE2 和 PE3 将只能通过 PE1 学到对方的路由。

说明：

此案例的配置重点有两个:

- 通过对不同 PE 上 VPN-target 属性的设置，控制总部与分公司之间的路由发布。
- 允许一次路由环路，使 PE 能够接收 CE 发送的含本 AS 号的路由更新。

HUB&SPOKE 组网中 PE1 上用于发布路由的 VPN-instance（VPN-instance3）的 vpn-target 不能跟 PE1 上用于引入路由的 VPN-instance（VPN-instance2）的任何一个 vpn-target 相同。

HUB&SPOKE 组网中 PE1 上用于发布路由的 VPN-instance 的 route-distinguisher rd2（100:2）不能跟各个 PE2、PE3 上相应 VPN-instance 的 route-distinguisher rd1（100:1）、rd4（100:4）中的任一个相同，而 rd1、rd4 可以相同也可以不同。

---

2. 组网图

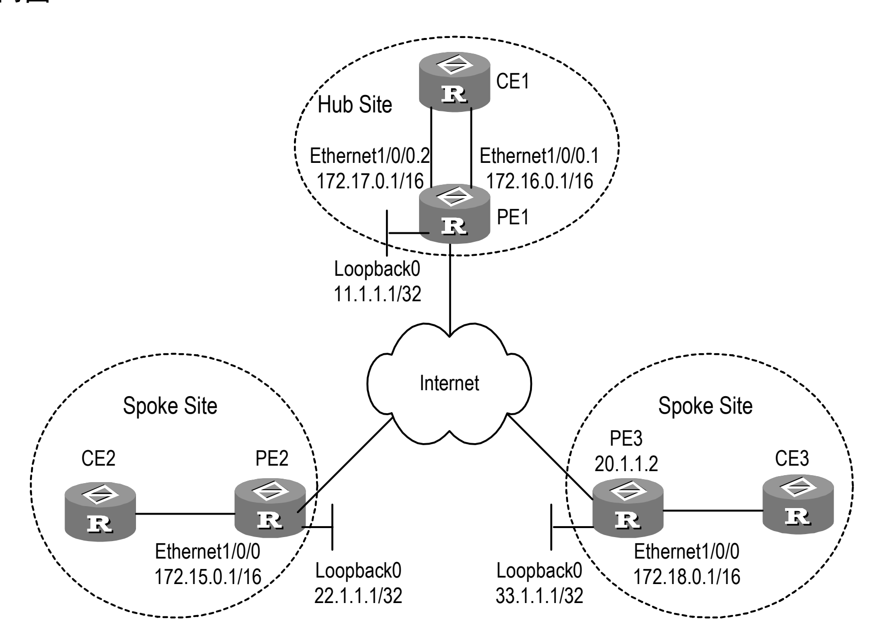


图3-10 Hub&Spoke 组网图

3. 配置步骤

说明：

本配置步骤部分省略了PE路由器之间的MPLS基本能力配置、及CE路由器的配置。这部分的内容可以参考 3.4.1 。

(1) 配置 PE1

在 PE1 上配置两个 VPN-instance，对从 PE2 和 PE3 接收的路由加上指定的 VPN-target 属性。

```
[PE1] ip vpn-instance vpn-instance2
[PE1-vpn- vpn-instance2] route-distinguisher 100:2
[PE1-vpn- vpn-instance2] vpn-target 100:1 import-extcommunity
[PE1-vpn-instance2] quit
[PE1] ip vpn-instance vpn-instance3
[PE1-vpn- vpn-instance3] route-distinguisher 100:3
[PE1-vpn- vpn-instance3] route-target 100:2 export-extcommunity
[PE1-vpn- vpn-instance3] quit
```

PE1 与 CE1 间建立 EBGP 邻居，将学到的 CE1 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族，并允许一次路由环路。

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn-instance2
[PE1-bgp-af-vpn-instance] group 17216 external
```

```
[PE1-bgp-af-vpn-instance] peer 172.16.1.1 group 17216 as-number 65002
[PE1-bgp-af-vpn-instance] peer 172.16.1.1 allow-as-loop 1
[PE1-bgp-af-vpn-instance] import-route static
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] ipv4-family vpn-instance vpn-instance3
[PE1-bgp-af-vpn-instance] group 17217 external
[PE1-bgp-af-vpn-instance] peer 172.17.1.1 group 17217 as-number 65002
[PE1-bgp-af-vpn-instance] peer 172.17.1.1 allow-as-loop 1
[PE1-bgp-af-vpn-instance] import-route static
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] quit
```

将 PE1 与 CE1 相连的接口绑定到不同的 VPN-instance。以太网子接口 Ethernet1/0/0.1 绑定到 vpn-instance2，以太网子接口 Ethernet1/0/0.2 绑定到 vpn-instance3。

```
[PE1] interface ethernet 1/0/0.1
[PE1-Ethernet1/0/0.1] ip binding vpn-instance vpn-instance2
[PE1-Ethernet1/0/0.1] ip address 172.16.0.1 255.255.0.0
[PE1-Ethernet1/0/0.1] quit
[PE1] interface ethernet 1/0/0.2
[PE1-Ethernet1/0/0.2] ip binding vpn-instance vpn-instance3
[PE1-Ethernet1/0/0.2] ip address 172.17.0.1 255.255.0.0
[PE1-Ethernet1/0/0.2] quit
```

配置 LoopBack 接口。

```
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 11.1.1.1 255.255.255.255
[PE1-LoopBack0] quit
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE1] bgp 100
[PE1-bgp] group 22
[PE1-bgp] peer 22.1.1.1 group 22
[PE1-bgp] peer 22.1.1.1 connect-interface loopback 0
[PE1-bgp] group 33
[PE1-bgp] peer 33.1.1.1 group 33
[PE1-bgp] peer 33.1.1.1 connect-interface loopback 0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer 22 enable
[PE1-bgp-af-vpn] peer 22.1.1.1 group 22
[PE1-bgp-af-vpn] peer 33 enable
```

```
[PE1-bgp-af-vpn] peer 33.1.1.1 group 33
[PE1-bgp-af-vpn] quit
```

(2) 配置 PE2

在 PE2 上创建 VPN-instance，允许引入 VPN-target 属性为 100:2 的 VPN 路由，发布的 VPN 路由的 VPN-target 属性为 100:1。

```
[PE2] ip vpn-instance vpn-instance1
[PE2-vpn-vpn-instance1] route-distinguisher 100:1
[PE2-vpn-vpn-instance1] route-target 100:1 export-extcommunity
[PE2-vpn-vpn-instance1] route-target 100:2 import-extcommunity
[PE2-vpn-vpn-instance1] quit
```

PE2 与 CE2 间建立 EBGP 邻居，将学到的 CE2 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn-instance1
[PE2-bgp-af-vpn-instance] group 172 external
[PE2-bgp-af-vpn-instance] peer 172.15.1.1 group 172 as-number 65001
[PE2-bgp-af-vpn-instance] import-route static
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] quit
[PE2-bgp] quit
```

将 PE2 与 CE2 相连的接口绑定到 VPN-instance。

```
[PE2] interface ethernet 1/0/0
[PE2-Ethernet1/0/0] ip binding vpn-instacne vpn-instance1
[PE2-Ethernet1/0/0] ip address 172.15.0.1 255.255.0.0
[PE2-Ethernet1/0/0] quit
```

配置 LoopBack 接口。

```
[PE2] interface loopback 0
[PE2-LoopBack0] ip address 22.1.1.1 255.255.255.255
[PE2-LoopBack0] quit
```

在 PE2 与 PE1 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE2] bgp 100
[PE2] group 11
[PE2-bgp] peer 11.1.1.1 group 11
[PE2-bgp] peer 11.1.1.1 connect-interface loopback 0
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpn] peer 11 enable
[PE2-bgp-af-vpn] peer 11.1.1.1 goup 11
[PE2-bgp-af-vpn] quit
```

```
[PE2-bgp] quit
```

(3) 配置 PE3

在 PE3 上创建 VPN-instance，允许引入 VPN-target 属性为 100:2 的 VPN 路由，发布的 VPN 路由的 VPN-target 属性为 100:1。

```
[PE3] ip vpn-instance vpn-instance2
[PE3-vpn-vpn-instance2] route-distinguisher 100:4
[PE3-vpn-vpn-instance2] route-target 100:1 export-extcommunity
[PE3-vpn-vpn-instance2] route-target 100:2 import-extcommunity
[PE3-vpn-vpn-instance2] quit
```

PE3 与 CE3 间建立 EBGP 邻居，将学到的 CE3 内部 VPN 路由引入 MBGP 的 VPN-instance 地址族。

```
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpn-instance1
[PE3-bgp-af-vpn-instance] group 172 external
[PE3-bgp-af-vpn-instance] peer 172.18.1.1 group 172 as-number 65001
[PE3-bgp-af-vpn-instance] import-route static
[PE3-bgp-af-vpn-instance] import-route direct
[PE3-bgp-af-vpn-instance] quit
[PE3-bgp] quit
```

将 PE3 与 CE3 相连的接口绑定到 VPN-instance。

```
[PE3] interface ethernet 1/0/0
[PE3-Ethernet1/0/0] ip binding vpn-instance vpn-instance2
[PE3-Ethernet1/0/0] ip address 172.18.0.1 255.255.0.0
[PE3-Ethernet1/0/0] quit
```

配置 LoopBack 接口。

```
[PE3] interface loopback 0
[PE3-LoopBack0] ip address 33.1.1.1 255.255.255.255
[PE3-LoopBack0] quit
```

在 PE3 与 PE1 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE3] bgp 100
[PE3-bgp] group 11
[PE3-bgp] peer 11.1.1.1 group 11
[PE3-bgp] peer 11.1.1.1 connect-interface loopback 0
[PE3-bgp] ipv4-family vpnv4
[PE3-bgp-af-vpn] peer 11 enable
[PE3-bgp-af-vpn] peer 11.1.1.1 group 11
[PE3-bgp-af-vpn] quit
[PE3-bgp] quit
```

3.4.5 CE 双归属组网举例

1. 组网需求

在对网络健壮性要求较高的应用中，可以采用 CE 双归属方式组网。

在下面的组网图中，CE1 和 CE2 分别与 PE1 和 PE2 设备相连，实现双归属；三个 PE 设备也两两相连，组成备份链路。CE3 和 CE4 不使用双归属，只与一个 PE 设备相连。

CE1 与 CE3 属于同一个 VPN；CE2 与 CE4 属于同一个 VPN。不同的 VPN 之间不能互通。

2. 组网图

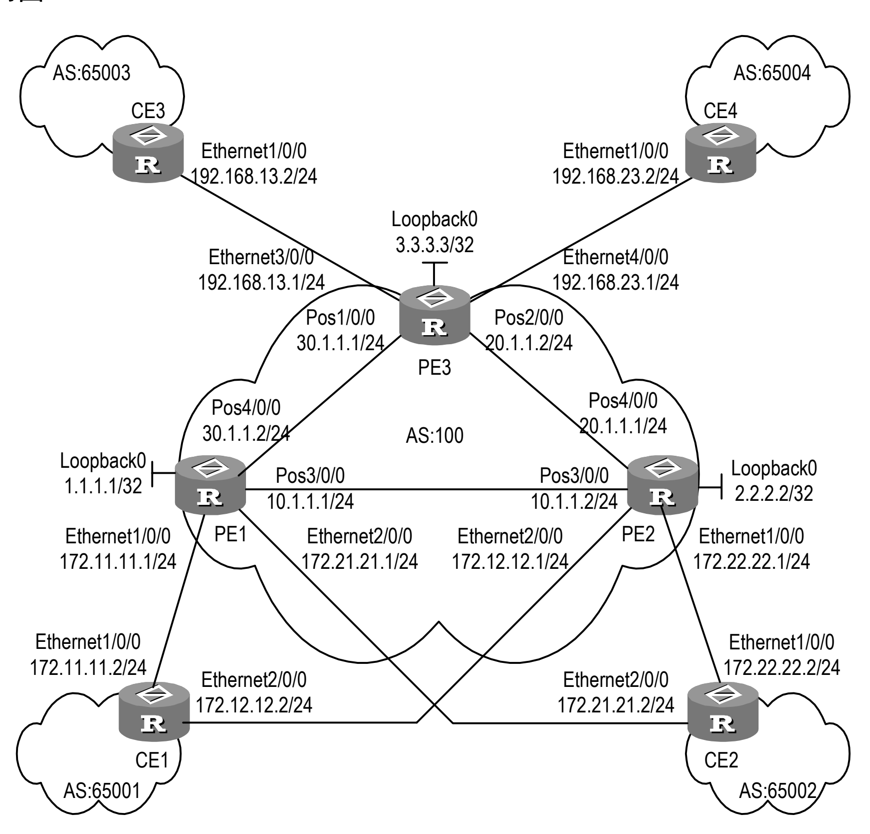


图3-11 CE 双归属组网图

3. 配置步骤

说明：

本配置步骤省略了CE路由器的配置，CE的配置可参考 3.4.1 。

(1) 配置 PE1

在 PE1 上为 CE1 和 CE2 分别创建 VPN-instance1.1 和 1.2，并配置不同的 VPN-target 属性。

```
[PE1] ip vpn-instance vpn-instance1.1
[PE1-vpn- vpn-instance1.1] route-distinguisher 1.1.1.1:1
[PE1-vpn- vpn-instance1.1] route-target 1.1.1.1:1
[PE1-vpn- vpn-instance1.1] vpn-target 2.2.2.2:1 import-extcommunity
[PE1-vpn- vpn-instance1.1] vpn-target 3.3.3.3:1 import-extcommunity
[PE1-vpn- vpn-instance1.1] quit
[PE1] ip vpn-instance vpn-instance1.2
[PE1-vpn- vpn-instance1.2] route-distinguisher 1.1.1.1:2
[PE1-vpn- vpn-instance1.2] route-target 1.1.1.1:2
[PE1-vpn- vpn-instance1.2] vpn-target 2.2.2.2:2 import-extcommunity
[PE1-vpn- vpn-instance1.2] vpn-target 3.3.3.3:2 import-extcommunity
[PE1-vpn- vpn-instance1.2] quit
```

在 PE1 与 CE1 间建立 EBGP 邻居，将 CE1 内部 VPN 路由引入 VPN-instance1.1。

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpn-instance1.1
[PE1-bgp-af-vpn-instance] group 17211 external
[PE1-bgp-af-vpn-instance] peer 172.11.11.2 group 17211 as-number 65001
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] import-route static
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] quit
```

在 PE1 与 CE2 间建立 EBGP 邻居，将 CE2 内部 VPN 路由引入 VPN-instance1.2。

```
[PE1-bgp] group 17221
[PE1-bgp] ipv4-family vpn-instance vpn-instance1.2
[PE1-bgp-af-vpn-instance] peer 172.21.21.2 group 17221 as-number 65002
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] import-route static
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] quit
```

将 PE1 与 CE1 相连的接口绑定到 VPN-instance1.1；将 PE1 与 CE2 相连的接口绑定到 VPN-instance1.2

```
[PE1] interface ethernet 1/0/0
[PE1-Ethernet1/0/0] ip binding vpn-instance vpn-instance1.1
[PE1-Ethernet1/0/0] ip address 172.11.11.1 255.255.255.0
[PE1-Ethernet1/0/0] quit
[PE1] interface ethernet 2/0/0
[PE1-Ethernet2/0/0] ip binding vpn-instance vpn-instance1.2
[PE1-Ethernet2/0/0] ip address 172.21.21.1 255.255.255.0
```

```
[PE1-Ethernet2/0/0] quit
```

配置 LoopBack 接口。

```
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 1.1.1.1 255.255.255.255
[PE1-LoopBack0] quit
```

配置 MPLS 基本能力，并在 PE1 与 PE2、PE3 相连的接口上使能 LDP。

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1] mpls ldp
[PE1] interface Serial 3/0/0
[PE1-Serial3/0/0] mpls
[PE1-Serial3/0/0] mpls ldp enable
[PE1-Serial3/0/0] ip address 10.1.1.1 255.255.255.0
[PE1-Serial3/0/0] interface Serial 4/0/0
[PE1-Serial4/0/0] mpls
[PE1-Serial4/0/0] mpls ldp enable
[PE1-Serial4/0/0] ip address 30.1.1.2 255.255.255.0
[PE1-Serial4/0/0] quit
```

在 PE1 与 P2、PE3 相连的接口及环回接口上启用 OSPF，实现 PE 内部的互通。

```
[PE1] router id 1.1.1.1
[PE1] ospf
[PE1-ospf-1] area 0
[PE1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[PE1-ospf-1-area-0.0.0.0] network 30.1.1.2 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] network 10.1.1.1 0.0.0.255
[PE1-ospf-1-area-0.0.0.0] quit
[PE1-ospf-1] quit
```

在 PE 与 PE 之间建立 MP-IBGP 邻居，进行 PE 内部的 VPN 路由信息交换。并在 VPNv4 地址族视图下激活 MP-IBGP 对等体。

```
[PE1] bgp 100
[PE1-bgp] group 2
[PE1-bgp] peer 2.2.2.2 group 2
[PE1-bgp] peer 2.2.2.2 connect-interface loopback 0
[PE1-bgp] group 3
[PE1-bgp] peer 3.3.3.3 group 3
[PE1-bgp] peer 3.3.3.3 connect-interface loopback 0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer 2 enable
[PE1-bgp-af-vpn] peer 2.2.2.2 group 2
[PE1-bgp-af-vpn] peer 3 enable
```

```
[PE1-bgp-af-vpn] peer 3.3.3.3 group 3
[PE1-bgp-af-vpn] quit
```

(2) 配置 PE2

📖 **说明：**

PE2 与 PE1 的配置基本相同，以下仅对 VPN-instance 的配置进行说明，其它配置不再赘述。

在 PE2 上为 CE1 和 CE2 分别创建 VPN-instance2.1 和 2.2，并配置不同的 VPN-target 属性。

```
[PE2] ip vpn-instance vpn-instance2.1
[PE2-vpn- vpn-instance2.1] route-distinguisher 2.2.2.2:1
[PE2-vpn- vpn-instance2.1] vpn-target 2.2.2.2:1
[PE2-vpn- vpn-instance2.1] vpn-target 1.1.1.1:1 import-extcommunity
[PE2-vpn- vpn-instance2.1e] vpn-target 3.3.3.3:1 import-extcommunity
[PE2-vpn- vpn-instance2.1] quit
[PE2] ip vpn-instance vpn-instance2.2
[PE2-vpn- vpn-instance2.2] route-distinguisher 2.2.2.2:2
[PE2-vpn- vpn-instance2.2] vpn-target 2.2.2.2:2
[PE2-vpn- vpn-instance2.2] vpn-target 1.1.1.1:2 import-extcommunity
[PE2-vpn- vpn-instance2.2e] vpn-target 3.3.3.3:2 import-extcommunity
[PE2-vpn- vpn-instance2.2] quit
```

在 PE2 与 CE1 间建立 EBGP 邻居，将 CE1 内部 VPN 路由引入 VPN-instance2.1。

```
[PE2] bgp 100
[PE2-bgp] ipv4-family vpn-instance vpn-instance2.1
[PE2-bgp-af-vpn-instance] group 17212 external
[PE2-bgp-af-vpn-instance] peer 172.12.12.2 group 17212 as-number 65001
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] import-route static
[PE2-bgp-af-vpn-instance] quit
```

在 PE2 与 CE2 间建立 EBGP 邻居，将 CE2 内部 VPN 路由引入 VPN-instance2.2。

```
[PE2-bgp] ipv4-family vpn-instance vpn-instance2.2
[PE2-bgp-af-vpn-instance] group 17222 external
[PE2-bgp-af-vpn-instance] peer 172.22.22.2 group 17222 as-number 65002
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] import-route static
[PE2-bgp-af-vpn-instance] quit
[PE2-bgp] quit
```

将 PE2 与 CE1 相连的接口绑定到 VPN-instance2.1；将 PE2 与 CE2 相连的接口绑定到 VPN-instance2.2

```
[PE2] interface ethernet 2/0/0
[PE2-Ethernet2/0/0] ip binding vpn-instance vpn-instance2.1
[PE2-Ethernet2/0/0] ip address 172.12.12.1 255.255.255.0
[PE2-Ethernet2/0/0] quit
[PE2] interface ethernet 1/0/0
[PE2-Ethernet1/0/0] ip binding vpn-instance vpn-instance2.2
[PE2-Ethernet1/0/0] ip address 172.22.22.1 255.255.255.0
[PE2-Ethernet1/0/0] quit
```

(3) 配置 PE3

📖 **说明：**

以下仅对 PE3 的 VPN-instance 配置进行说明，其它配置与 PE1 和 PE2 类似，不再赘述。

在 PE3 上为 CE3 和 CE4 分别创建 VPN-instance3.1 和 3.2，并配置不同的 VPN-target 属性。

```
[PE3] ip vpn-instance vpn-instance3.1
[PE3-vpn- vpn-instance3.1] route-distinguisher 3.3.3.3:1
[PE3-vpn- vpn-instance3.1] vpn-target 3.3.3.3:1
[PE3-vpn- vpn-instance3.1] vpn-target 1.1.1.1:1 import-extcommunity
[PE3-vpn- vpn-instance3.1] vpn-target 2.2.2.2:1 import-extcommunity
[PE3-vpn- vpn-instance3.1] quit
[PE3] ip vpn-instance vpn-instance3.2
[PE3-vpn-instance] route-distinguisher 3.3.3.3:2
[PE3-vpn-instance] vpn-target 3.3.3.3:2
[PE3-vpn-instance] vpn-target 1.1.1.1:2 import-extcommunity
[PE3-vpn-instance] vpn-target 2.2.2.2:2 import-extcommunity
[PE3-vpn-instance] quit
```

在 PE3 与 CE3 间建立 EBGP 邻居，将 CE3 内部 VPN 路由引入 VPN-instance3.1。

```
[PE3] bgp 100
[PE3-bgp] ipv4-family vpn-instance vpn-instance3.1
[PE3-bgp-af-vpn-instance] group 192 external
[PE3-bgp-af-vpn-instance] peer 192.168.13.2 group 192 as-number 65003
[PE3-bgp-af-vpn-instance] import-route direct
[PE3-bgp-af-vpn-instance] import-route static
[PE3-bgp-af-vpn-instance] quit
[PE3-bgp] quit
```

在 PE3 与 CE4 间建立 EBGP 邻居，将 CE4 内部 VPN 路由引入 VPN-instance3.2。

```
[PE3-bgp] ipv4-family vpn-instance vpn-instance3.2
[PE3-bgp-af-vpn-instance] group 232 external
```

```
[PE3-bgp-af-vpn-instance] peer 192.168.23.2 group 232 as-number 65004
[PE3-bgp-af-vpn-instance] import-route direct
[PE3-bgp-af-vpn-instance] import-route static
[PE3-bgp-af-vpn-instance] quit
[PE3-bgp] quit
```

将 PE3 与 CE3 相连的接口绑定到 VPN-instance3.1；将 PE3 与 CE2 相连的接口绑定到 VPN-instance3.2

```
[PE3] interface ethernet 3/0/0
[PE3-Ethernet3/0/0] ip binding vpn-instance vpn-instance3.1
[PE3-Ethernet3/0/0] ip address 192.168.13.1 255.255.255.0
[PE3-Ethernet3/0/0] quit
[PE3] interface ethernet 4/0/0
[PE3-Ethernet4/0/0] ip binding vpn-instance vpn-instance3.2
[PE3-Ethernet4/0/0] ip address 192.168.23.1 255.255.255.0
[PE3-Ethernet4/0/0] quit
```

3.4.6 多角色主机组网举例

1. 组网需求

如下图所示，某公司通过 BGP/MPLS VPN 来提供 VPN 服务。

PE1 与 CE1 相连的 Serial0/0/0 接口绑定到 VPN1，PE2 与 CE2 相连的接口 Serial0/0/0 绑定到 VPN2。

主机 PC1 通过 CE1 接入，PC1 的 IP 地址为 100.1.1.2，配置完成后，它将可以访问 VPN1 和 VPN2。

说明：

此案例的配置要点在于:

通过配置静态路由和路由策略，使 PC1 访问不同 VPN 的报文在不同的 VPN-instance 中查找路由。

---

2. 组网图

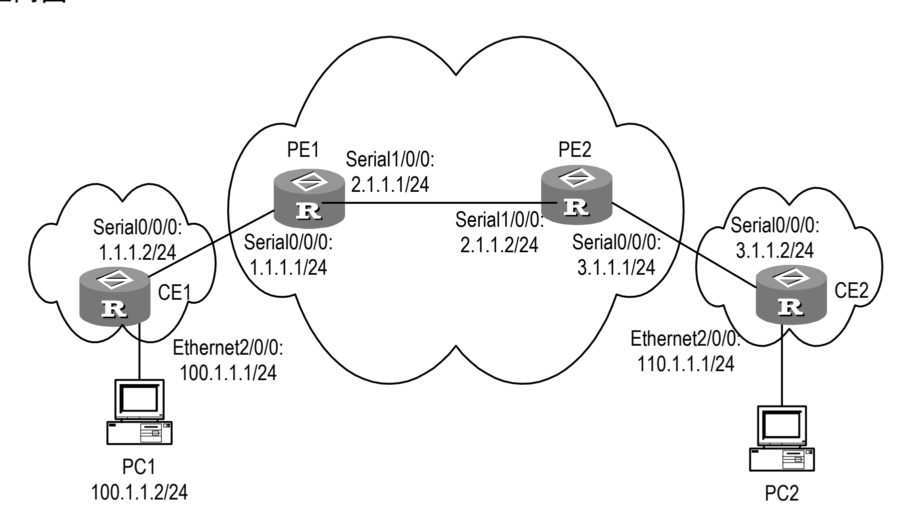


图3-12 多角色主机组网图

3. 配置步骤

(1) 配置 CE1

在 CE1 上配置一条指向 PE1 的缺省路由。

```
[CE1] ip route-static 0.0.0.0 0 1.1.1.1
```

(2) 配置 PE1

在 PE1 上为 VPN1 和 VPN2 分别创建 VPN-instance，并配置不同的 VPN-target 属性。

```
[PE1] ip vpn-instance vpn1
[PE1-vpn- vpn1] route-distinguisher 100:1
[PE1-vpn- vpn1] vpn-target 100:1 both
[PE1-vpn- vpn1] quit
[PE1] ip vpn-instance vpn2
[PE1-vpn- vpn2] route-distinguisher 100:2
[PE1-vpn- vpn2] vpn-target 100:2 both
[PE1-vpn- vpn2] quit
```

将 PE1 与 CE1 相连的接口绑定到 VPN1。

```
[PE1] interface serial0/0/0
[PE1-Serial0/0/0] ip binding vpn-instance vpn1
[PE1-Serial0/0/0] ip address 1.1.1.1 255.255.255.0
[PE1-Serial0/0/0] quit
```

配置静态路由，使 PC1 访问 VPN2 的返回报文能够在 PE1 的 VPN-instance vpn1 中找到正确的路由，回到 PC1（相应的应在 PE2 中配置 VPN2 引入直连路由，以便通过 MBGP 向整个 VPN2 发布这条路由）。

```
[PE1] ip route-static vpn-instance vpn2 100.1.0.0 16 vpn-instance vpn1 1.1.1.2
```

配置策略路由，对于 PC1 发出的报文，可以同时在 vpn1 和 vpn2 中查找私网路由并转发。

```
[PE1] acl 3101
[PE1-acl-adv-3101] rule 0 permit ip vpn-instance vpna source 100.1.1.2 0
[PE1-acl-adv-3101] quit
[PE1] route-policy aaa permit node 10
[PE1-route-policy] if-match acl 3101
[PE1-route-policy] apply access-vpn vpn-instance vpn2
[PE1-route-policy] quit
```

在接口 Serial0/0/0 上应用定义的策略路由。

```
[PE1] interface serial0/0/0
[PE1-Serial0/0/0] ip policy route-policy aaa
```

其他配置请参考前面的例子。

3.4.7 跨域的 BGP/MPLS VPN 组网案例

1. 组网需求

某 VPN 用户在城市 A 和城市 B 都有站点。由于地域方面的原因，城市 A 的站点接入城市 A 的服务提供商的 MPLS/VPN 网络，自治域号为 AS100；而城市 B 的站点接入城市 B 的服务提供商 MPLS/VPN 网络，自治域号为 AS200。用户的 VPN 网络跨两个 AS 网络。CE1、CE2 属于一个 VPNA，CE3、CE4 属于一个 VPNB。

2. 组网图

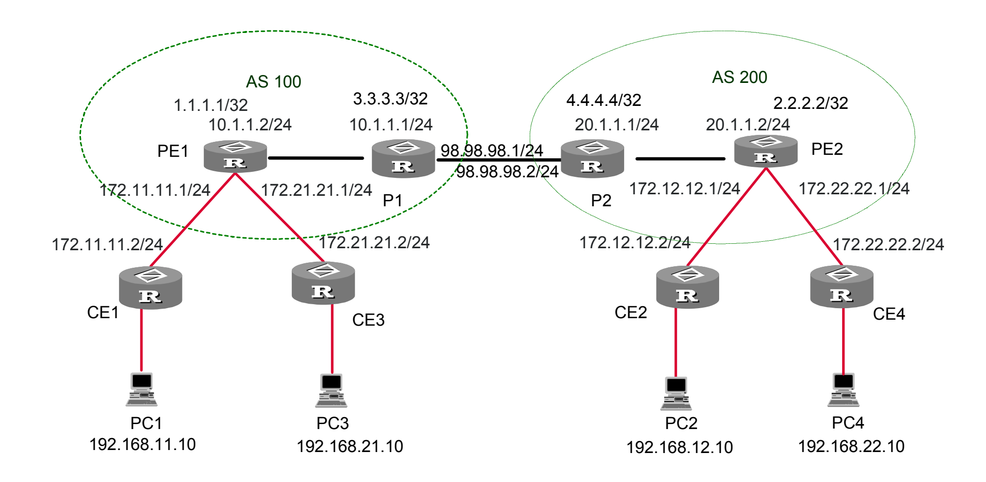


图3-13 ASBR 组网图

3. 配置步骤

(1) 配置 PE1

使能 MPLS 及 LDP。

```
[PE1] mpls lsr-id 1.1.1.1
[PE1] mpls
[PE1] mpls ldp
```

配置连接 CE 的 VLAN 子接口。

```
[PE1] interface ethernet1/0/0.1
[PE1-ethernet1/0/0.1] vlan-type dot1q vid 10
[PE1-ethernet1/0/0.1] mpls
[PE1-ethernet1/0/0.1] mpls ldp enable
[PE1-ethernet1/0/0.1] ip address 172.11.11.1 255.255.255.0
[PE1] interface ethernet1/0/1.1
[PE1-ethernet1/0/1.1] vlan-type dot1q vid 20
[PE1-ethernet1/0/1.1] mpls
[PE1-ethernet1/0/1.1] mpls ldp enable
[PE1-ethernet1/0/1.1] ip address 172.21.21.1 255.255.255.0
```

配置 loopback 接口。

```
[PE1] interface loopback 0
[PE1-LoopBack0] ip address 1.1.1.1 255.255.255.255
```

配置 VPN 实例。

```
[PE1] ip vpn-instance vpna
[PE1-vpn-vpna] route-distinguisher 100:1
[PE1-vpn-vpna] vpn-target 100:1 both
[PE1] ip vpn-instance vpnb
[PE1-vpn-vpnb] route-distinguisher 100:2
[PE1-vpn-vpnb] vpn-target 100:2 both
```

配置 Serial 口。

```
[PE1] interface s0/0/0
[PE1-s0/0/0] ip address 10.1.1.2 255.255.255.0
```

配置 VLAN 子接口与 **vpn-instance** 的绑定。

```
[PE1] interface ethernet1/0/0.1
[PE1-ethernet1/0/0.1] ip binding vpn-instance vpna
[PE1] interface ethernet1/0/1.1
[PE1-ethernet1/0/1.1] ip binding vpn-instance vpnb
```

配置 PE-CE 之间运行 EBGP。

```
[PE1] bgp 100
[PE1-bgp] ipv4-family vpn-instance vpna
[PE1-bgp-af-vpn-instance] group 172-11 external
[PE1-bgp-af-vpn-instance] peer 172.11.11.2 group 172-11 as-number 65011
[PE1-bgp-af-vpn-instance] peer 172.11.11.2 next-hop-local
```

```
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
[PE1-bgp] ipv4-family vpn-instance vpnb
[PE1-bgp-af-vpn-instance] group 172-21 external
[PE1-bgp-af-vpn-instance] peer 172.21.21.2 group 172-21 as-number 65021
[PE1-bgp-af-vpn-instance] peer 172.21.21.2 next-hop-local
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] quit
```

配置 PE-PE 之间运行 IBGP。

```
[PE1-bgp] group 3 external
[PE1-bgp] peer 3.3.3.3 group 3
[PE1-bgp] peer 3.3.3.3 connect-interface loopback0
[PE1-bgp] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer 3 enable
[PE1-bgp-af-vpn] peer 3.3.3.3 group 3
```

(2) 配置 PE2

MPLS 配置

```
[PE2] mpls lsr-id 2.2.2.2
[PE2] mpls
[PE2] mpls ldp
```

配置连接 CE 的 VLAN 子接口。

```
[PE2] interface ethernet1/0/0.1
[PE2-ethernet1/0/0.1] vlan-type dot1q vid 10
[PE2-ethernet1/0/0.1] mpls
[PE2-ethernet1/0/0.1] mpls ldp enable
[PE2-ethernet1/0/0.1] ip address 172.12.12.1 255.255.255.0
[PE2] interface ethernet1/0/1.1
[PE2-ethernet1/0/1.1] vlan-type dot1q vid 20
[PE2-ethernet1/0/1.1] mpls
[PE2-ethernet1/0/1.1] mpls ldp enable
[PE2-ethernet1/0/1.1] ip address 172.22.22.1 255.255.255.0
```

配置 loopback 接口。

```
[PE2] interface loopback 0
[PE2-LoopBack0] ip address 2.2.2.2 255.255.255.255
```

配置 VPN 实例

```
[PE2] ip vpn-instance vpna
[PE2-vpn-vpna] route-distinguisher 200:1
[PE2-vpn-vpna] vpn-target 100:1 both
[PE2] ip vpn-instance vpnb
```

```
[PE2-vpn-vpnb] route-distinguisher 200:2
[PE2-vpn-vpnb] vpn-target 100:2 both
```

配置 VLAN 子接口与 vpn-instance 的绑定。

```
[PE2] interface ethernet1/0/0.1
[PE2-ethernet1/0/0.1] ip binding vpn-instance vpna
[PE2] interface ethernet1/0/1.1
[PE2-ethernet1/0/1.1] ip binding vpn-instance vpnb
```

在 PE-CE 之间配置运行 EBGP。

```
[PE2] bgp 200
[PE2-bgp] ipv4-family vpn-instance vpna
[PE2-bgp-af-vpn-instance] peer 172.12.12.2 group 172-12 as-number 65012
[PE2-bgp-af-vpn-instance] group 172-12 external
[PE2-bgp-af-vpn-instance] peer 172.12.12.2 next-hop-local
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] group 172-22 external
[PE2-bgp-af-vpn-instance] quit
[PE2-bgp] ipv4-family vpn-instance vpnb
[PE2-bgp-af-vpn-instance] peer 172.22.22.2 group 172-22 as-number 65022
[PE2-bgp-af-vpn-instance] peer 172.22.22.2 next-hop-local
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] quit
```

在 PE-PE 之间配置运行 IBGP。

```
[PE2-bgp] group 4
[PE2-bgp] peer 4.4.4.4 group 4
[PE2-bgp] peer 4.4.4.4 connect-interface loopback0
[PE2-bgp] ipv4-family vpnv4
[PE2-bgp-af-vpn] peer 4 enable
[PE2-bgp-af-vpn] peer 4.4.4.4 group 4
```

(3) 配置 P1（P2 配置与此相似）

MPLS 基本功能配置

```
[P1] mpls lsr-id 3.3.3.3
[P1] mpls
[P1] mpls ldp
```

配置 loopback 0 接口。

```
[P1] interface loopback 0
[P1-LoopBack0] ip address 3.3.3.3 255.255.255.255
```

配置 Serial 口。

```
[P1] interface s0/0/0
[P1- s0/0/0] mpls
```

```
[P1-s0/0/0] mpls ldp enable
[P1-s0/0/0 ip address 10.1.1.1 255.255.255.0
[P1] interface s1/0/0
[P1-s1/0/0] mpls
[P1-s1/0/0] mpls ldp enable
[P1-s1/0/0] ip address 98.98.98.1 255.255.255.0
```

配置 IBGP 邻居和 EBGP 的邻居。

```
[P1] bgp 100
[P1-bgp] group 1 internal
[P1-bgp] peer 1.1.1.1 group 1
[P1-bgp] peer 1.1.1.1 connect-interface loopback0
[P1-bgp] peer 1.1.1.1 next-hop-local
[P1-bgp] group 98 external
[P1-bgp] peer 98.98.98.2 group 98
[P1-bgp] ipv4-family vpnv4
[P1-bgp-af-vpn] peer 1 enable
[P1-bgp-af-vpn] peer 1.1.1.1 group 1
[P1-bgp-af-vpn] peer 98 enable
[P1-bgp-af-vpn] peer 98.98.98.2 group 98
[P1-bgp-af-vpn] undo policy vpn-target
```

3.4.8 分层式 BGP/MPLS VPN 组网案例

1. 组网需求

对于层次化比较明显的 VPN 网络，比如一个包括省骨干和地市的 MPLS VPN 网络，在网络拓扑比较大的情况下，如果将省网和地市网组到一个 MPLS VPN 网络中，将对整个网络中的设备的性能要求比较高。如果我们将这个 MPLS VPN 网络分成上下两个 MPLS VPN 网络的的话，比如省网和地市网络，就可以解决对设备性能要求较高的问题。

SPE 作为省网的一个 PE 设备，它的下面要接一个地市的 MPLS VPN 网络，UPE 作为下层地市网络的一个 PE 设备，它要最终接入 VPN 客户，一般为低端路由器。

2. 组网图

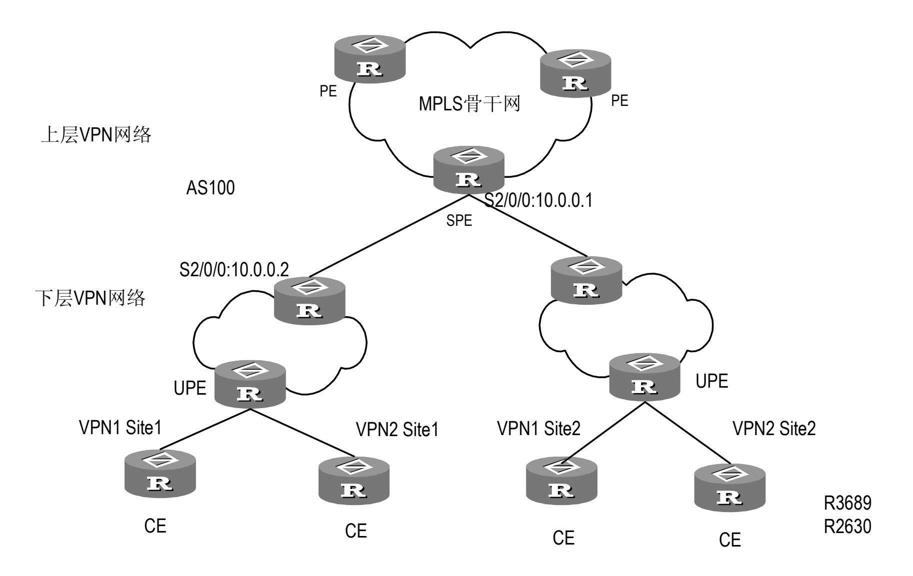


图3-14 分层式 BGP/MPLS VPN 组网图

3. 配置步骤

📖 说明：

本案例仅给出体现分层式 BGP/MPLS VPN 中 PE 的相关配置。

(1) SPE 上的配置

配置 MPLS 基本能力

```
[SPE] mpls lsr-id 1.0.0.2
[SPE] mpls
[SPE] mpls ldp
```

VPN-INSTANCE 配置

```
[SPE] ip vpn-instance vpn1
[SPE-vpn-vpn1] route-distinguisher 100:1
[SPE-vpn-vpn1] vpn-target 100:1 both
```

接口配置（对 PE 路由器，配置 LOOPBACK 0 接口地址时，必须使用 32 位掩码的主机地址）

```
[SPE] interface serial2/0/0
[SPE-Serial2/0/0] ip address 10.0.0.1 255.0.0.0
[SPE-Serial2/0/0] mpls
[SPE-Serial2/0/0] mpls ldp enable
[SPE-Serial2/0/0] ip binding vpn-instance vpn1
[SPE] interface loopback0
[SPE-LoopBack 0] ip address 1.0.0.2 255.255.255.255
```

BGP 配置

```
[SPE] bgp 100
[SPE-bgp] ipv4-family vpn-instance vpna
[SPE--bgp-af-vpn-instance] group 1
[SPE--bgp-af-vpn-instance] peer 1.0.0.1 group 1 as-number 100
[SPE--bgp-af-vpn-instance] peer 1.0.0.1 upe
[SPE--bgp-af-vpn-instance] peer 1.0.0.1 defaulte-originate vpn1
[SPE--bgp-af-vpn-instance] peer 1.0.0.1 next-hop-local
[SPE--bgp-af-vpn-instance] quit
[SPE -bgp] quit
```

配置 OSPF

```
[SPE] ospf
[SPE-ospf-1] area 0
[SPE-ospf-1-area-0.0.0.0] network 0.0.0.0 0.0.0.0
[SPE-ospf-1-area-0.0.0.0] network 202.100.1.1 0.0.0.0
[SPE-ospf-1-area-0.0.0.0] import-route direct
```

(2) UPE 上的配置

配置 MPLS 基本能力

```
[UPE] mpls lsr-id 1.0.0.1
[UPE] mpls
[UPE] mpls ldp
```

VPN-INSTANCE 配置

```
[UPE] ip vpn-instance vpn1
[UPE-vpn-vpn1] route-distinguisher 100:1
[UPE-vpn-vpn1] vpn-target 100:1 both
```

接口配置

```
[UPE] interface serial2/0/0
[UPE-Serial2/0/0] ip address 10.0.0.2 255.0.0.0
[UPE-Serial2/0/0] mpls
[UPE-Serial2/0/0] mpls ldp enable
[UPE-Serial2/0/0] ip binding vpn-instance vpn1
[UPE] interface loopback0
[UPE-LoopBack 0] ip address 1.0.0.1 255.255.255.255
```

BGP 配置

```
[UPE] bgp 100
[UPE-bgp] group 1
[UPE-bgp] peer 1.0.0.2 group 1
[UPE-bgp] peer 1.0.0.2 connect-interface loopback0
[UPE-bgp] ipv4-family vpnv4
```

```
[UPE-bgp-af-vpn] peer 1 enable
[UPE-bgp-af-vpn] peer 1.0.0.2 group 1
```

配置 OSPF

```
[UPE] ospf
[UPE-ospf-1] area 0
[UPE-ospf-1-area-0.0.0.0] network 0.0.0.0 0.0.0.0
[UPE-ospf-1-area-0.0.0.0] import-route direct
[UPE-ospf-1-area-0.0.0.0] quit
```

3.4.9 OSPF 多实例 sham link 配置举例

1. 组网需求

如下图所示，一个公司通过 Quidway 路由器的 OSPF 多实例功能连接到广域网，其中 OSPF 绑定于 VPN1。PE 之间是 MPLS VPN 骨干网，PE 和 CE 之间运行 OSPF。在 PE1 和 PE2 之间配置一条 sham link，使得 CE1 和 CE2 之间的流量不会通过 CE1 和 CE2 之间直接相连的链路(backdoor link)。

2. 组网图

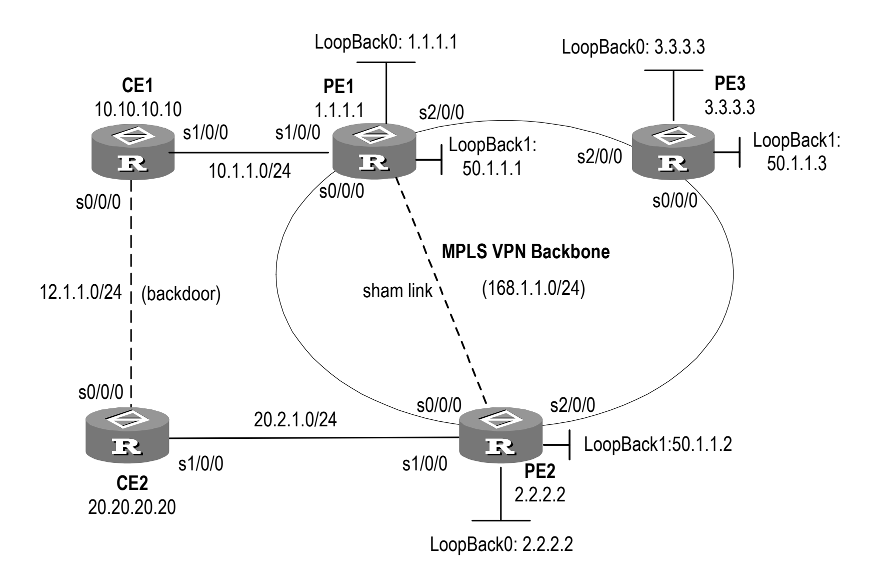


图3-15 OSPF 多实例配置案例组网图

3. 配置步骤

(1) 配置 PE1

使能 MPLS 及 LDP。

```
[PE1] mpls lsr-id 50.1.1.2
[PE1] mpls
[PE1] mpls ldp
```

配置 vpn-instance。

```
[PE1] ip vpn-instance VPN1
[PE1-vpn-VPN1] route-distinguisher 2:1
[PE1-vpn-VPN1] vpn-target 100:1 export-extcommunity
[PE1-vpn-VPN1] vpn-target 100:1 import-extcommunity
```

配置接口。

```
[PE1] interface Serial0/0/0
[PE1-Serial0/0/0] link-protocol ppp
[PE1-Serial0/0/0] ip address 168.1.12.1 255.255.255.0
[PE1-Serial0/0/0] ospf cost 1
[PE1-Serial0/0/0] mpls
[PE1-Serial0/0/0] mpls ldp enable
[PE1-Serial0/0/0] mpls ldp transport-ip interface
[PE1] interface Serial1/0/0
[PE1-Serial1/0/0] link-protocol ppp
[PE1-Serial1/0/0] ip binding vpn-instance VPN1
[PE1-Serial1/0/0] ip address 10.1.1.2 255.255.255.0
[PE1-Serial1/0/0] ospf cost 1
[PE1] interface Serial2/0/0
[PE1-Serial2/0/0] link-protocol ppp
[PE1-Serial2/0/0] ip address 168.1.13.1 255.255.255.0
[PE1-Serial2/0/0] ospf cost 1
[PE1-Serial2/0/0] mpls
[PE1-Serial2/0/0] mpls ldp enable
[PE1-Serial2/0/0] mpls ldp transport-ip interface
[PE1] interface loopback0
[PE1-LoopBack0] ip binding vpn-instance VPN1
[PE1-LoopBack0] ip address 1.1.1.1 255.255.255.255
[PE1] interface loopback1
[PE1-LoopBack1] ip address 50.1.1.1 255.255.255.255
```

配置 BGP peer。

```
[PE1] bgp 100
[PE1-bgp] undo synchronization
[PE1-bgp] group fc internal
[PE1-bgp] peer 2.2.2.2 group fc
[PE1-bgp] peer 2.2.2.2 connect-interface LoopBack1
[PE1-bgp] peer 3.3.3.3 group fc
[PE1-bgp] peer 3.3.3.3 connect-interface LoopBack1
```

配置 BGP 引入 OSPF 路由及直连路由。

```
[PE1-bgp] ipv4-family vpn-instance vpn1
[PE1-bgp-af-vpn-instance] import-route ospf 100
```

```
[PE1-bgp-af-vpn-instance] import-route ospf-ase 100
[PE1-bgp-af-vpn-instance] import-route ospf-nssa 100
[PE1-bgp-af-vpn-instance] import-route direct
[PE1-bgp-af-vpn-instance] undo synchronization
```

在 MBGP 下建立 peer 并激活。

```
[PE1-bgp-af-vpn] ipv4-family vpnv4
[PE1-bgp-af-vpn] peer fc enable
[PE1-bgp-af-vpn] peer fc advertise-community
[PE1-bgp-af-vpn] peer 2.2.2.2 group fc
[PE1-bgp-af-vpn] peer 3.3.3.3 group fc
```

绑定 OSPF 进程到 vpn-instance。

```
[PE1] ospf 100 router-id 1.1.1.1 vpn-instance VPN1
[PE1-ospf-100] import-route bgp
[PE1-ospf-100] area 0.0.0.1
[PE1-ospf-100-area-0.0.0.1] network 10.1.1.0  0.0.0.255
```

配置 sham link。

```
[PE1-ospf-100-area-0.0.0.1] sham-link 11.11.11.11 22.22.22.22
```

配置到 PE2、PE3 的静态路由。

```
[PE1] ip route-static 50.1.1.2 255.255.255.255 168.1.12.2
[PE1] ip route-static 50.1.1.3 255.255.255.255 168.1.13.3
```

(2) 配置 PE2

使能 MPLS 及 LDP。

```
[PE2] mpls lsr-id 50.0.0.1
[PE2] mpls
[PE2] mpls ldp
```

配置 vpn-instance VPN1。

```
[PE2] ip vpn-instance VPN1
[PE2-vpn-VPN1] route-distinguisher 2:1
[PE2-vpn-VPN1] vpn-target 100:1 export-extcommunity
[PE2-vpn-VPN1] vpn-target 100:1 import-extcommunity
```

配置接口。

```
[PE2] interface Serial0/0/0
[PE2-Serial0/0/0] link-protocol ppp
[PE2-Serial0/0/0] ip address 168.1.12.2 255.255.255.0
[PE2-Serial0/0/0] ospf cost 1
[PE2-Serial0/0/0] mpls
[PE2-Serial0/0/0] mpls ldp enable
[PE2-Serial0/0/0] mpls ldp transport-ip interface
```

```
[PE2] interface Serial1/0/0
[PE2-Serial1/0/0] link-protocol ppp
[PE2-Serial1/0/0] ip binding vpn-instance VPN1
[PE2-Serial1/0/0] ip address 20.1.1.2 255.255.255.0
[PE2-Serial1/0/0] ospf cost 1
[PE2] interface Serial2/0/0
[PE2-Serial2/0/0] link-protocol ppp
[PE2-Serial2/0/0] ip address 168.1.23.2 255.255.255.0
[PE2-Serial2/0/0] ospf cost 1
[PE2-Serial1/0/0] mpls
[PE2-Serial1/0/0] mpls ldp enable
[PE2-Serial1/0/0] mpls ldp transport-ip interface
[PE2] interface LoopBack0
[PE2- LoopBack0] ip binding vpn-instance VPN1
[PE2- LoopBack0] ip address 2.2.2.2 255.255.255.255
[PE2] interface LoopBack1
[PE2- LoopBack1] ip address 50.1.1.2 255.255.255.255
```

配置 BGP。

```
[PE2] bgp 100
[PE2-bgp] undo synchronization
[PE2-bgp] group fc internal
[PE2-bgp] peer 50.1.1.1 group fc
[PE2-bgp] peer 50.1.1.1 connect-interface LoopBack10
[PE2-bgp] peer 50.1.1.3 group fc
[PE2-bgp] peer 50.1.1.3 connect-interface LoopBack10
```

配置 vpn-instance 引入 OSPF 及直连路由。

```
[PE2-bgp] ipv4-family vpn-instance VPN1
[PE2-bgp-af-vpn-instance] import-route direct
[PE2-bgp-af-vpn-instance] import-route ospf-nssa 100
[PE2-bgp-af-vpn-instance] import-route ospf-ase 100
[PE2-bgp-af-vpn-instance] import-route ospf 100
[PE2-bgp-af-vpn-instance] undo synchronization
```

配置 MBGP 使能 peer。

```
[PE2-bgp-af-vpn] ipv4-family vpnv4
[PE2-bgp-af-vpn] peer fc enable
[PE2-bgp-af-vpn] peer fc advertise-community
[PE2-bgp-af-vpn] peer 50.1.1.1 group fc
[PE2-bgp-af-vpn] peer 50.1.1.3 group fc
```

配置 OSPF 引入 BGP 及直连路由。

```
[PE2] ospf 100 router-id 2.2.2.2 vpn-instance VPN1
```

```
[PE2-ospf-100] import-route bgp
[PE2-ospf-100] import-route static
[PE2-ospf-100] area 0.0.0.1
[PE2-ospf-100] network 20.1.1.0 0.0.0.255
```

配置 sham link。

```
[PE2-ospf-100] sham-link 2.2.2.2 1.1.1.1
```

配置到 PE1、PE3 的静态路由。

```
[PE2] ip route-static 50.1.1.1 255.255.255.255 168.1.12.1
[PE2] ip route-static 50.1.1.3 255.255.255.255 168.1.23.3
```

(3) 配置 CE1

配置接口。

```
[CE1] interface Serial0/0/0
[CE1-Serial0/0/0] link-protocol ppp
[CE1-Serial0/0/0] ip address 12.1.1.1 255.255.255.0
[CE1-Serial0/0/0] ospf cost 1
[CE1] interface Serial1/0/0
[CE1-Serial1/0/0] link-protocol ppp
[CE1-Serial1/0/0] ip address 10.1.1.1 255.255.255.0
[CE1-Serial1/0/0] ospf cost 1
```

配置 OSPF。

```
[CE1] ospf 100 router-id 10.10.10.129
[CE1-ospf-100] import-route direct
[CE1-ospf-100] area 0.0.0.1
[CE1-ospf-100] network 10.1.1.0 0.0.0.255
[CE1-ospf-100] network 12.1.1.0 0.0.0.255
```

(4) 配置 CE2

配置接口。

```
[CE2] interface Serial0/0/0
[CE2-Serial0/0/0] link-protocol ppp
[CE2-Serial0/0/0] ip address 12.1.1.2 255.255.255.0
[CE2-Serial0/0/0] ospf cost 1
[CE2] interface Serial1/0/0
[CE2-Serial1/0/0] link-protocol ppp
[CE2-Serial1/0/0] ip address 20.1.1.1 255.255.255.0
[CE2-Serial1/0/0] ospf cost 1
```

配置 OSPF。

```
[CE2] ospf 100 router-id 20.20.20.20
[CE2-ospf-100] area 0.0.0.1
[CE2-ospf-100] network 12.1.1.0 0.0.0.255
```

```
[CE2-ospf-100] network 20.1.1.0 0.0.0.255
```

3.4.10 OSPF 多实例 CE 配置举例

1. 组网需求

在一个 VPN 内部的 CE 路由器通过配置多个 vpn-instance 实现业务隔离。

2. 组网图

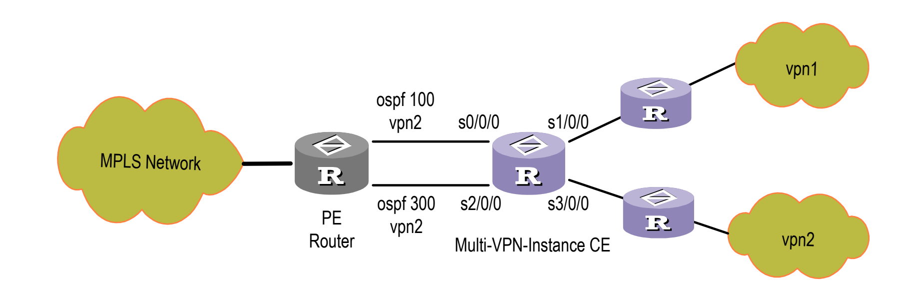


图3-16 OSPF 多实例 CE 配置举例

3. 配置步骤

(1) 配置 CE 路由器

配置实例 CE-VPN1

```
[CE] ip vpn-instance CE-VPN1
[CE-vpn-CE-VPN1] route-distinguisher 100:1
[CE-vpn-CE-VPN1] vpn-target 100:1 export-extcommunity
[CE-vpn-CE-VPN1] vpn-target 100:1 import-extcommunity
```

配置实例 CE-VPN2

```
[CE]ip vpn-instance CE-VPN2
[CE-vpn-CE-VPN2] route-distinguisher 200:1
[CE-vpn-CE-VPN2] vpn-target 200:1 export-extcommunity
[CE-vpn-CE-VPN2] vpn-target 200:1 import-extcommunity
```

配置 serial0/0/0

```
[CE] interface Serial0/0/0
[CE-Serial 0/0/0] link-protocol ppp
[CE-Serial 0/0/0] ip binding vpn-instance CE-VPN1
[CE-Serial 0/0/0] ip address 10.1.1.2 255.255.255.0
```

配置 serial1/0/0

```
[CE] interface Serial1/0/0
[CE-Serial1/0/0] link-protocol ppp
[CE-Serial1/0/0] ip binding vpn-instance CE-VPN1
[CE-Serial1/0/0] ip address 10.2.1.2 255.255.255.0
```

```
[CE-Serial1/0/0] ospf cost 100
```

配置 serial2/0/0

```
[CE] interface Serial2/0/0
[CE-Serial2/0/0] link-protocol ppp
[CE-Serial2/0/0] ip binding vpn-instance CE-VPN2
[CE-Serial2/0/0] ip address 20.1.1.2 255.255.255.0
```

配置 serial3/0/0

```
[CE] interface Serial3/0/0
[CE-Serial3/0/0] link-protocol ppp
[CE-Serial3/0/0] ip binding vpn-instance CE-VPN2
[CE-Serial3/0/0] ip address 20.2.1.2 255.255.255.0
```

配置 ospf 100

```
[CE] ospf 100 vpn-instance CE-VPN1
[CE-ospf-100] vpn-instance-capability simple
[CE-ospf-100] area 0.0.0.0
[CE-ospf-100-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[CE-ospf-100-area-0.0.0.0] network 10.2.1.0 0.0.0.255
```

配置 ospf 200

```
[CE] ospf 200 vpn-instance CE-VPN2
[CE-ospf-200] vpn-instance-capability simple
[CE-ospf-200] area 0.0.0.1
[CE-ospf-100-area-0.0.0.1] network 20.1.1.0 0.0.0.255
[CE-ospf-100-area-0.0.0.1] network 20.2.1.0 0.0.0.255
```

3.5 故障诊断与排错

3.5.1 多角色主机应用故障诊断与排错

故障之一：在多角色主机应用过程中，没有进行正确的策略路由转发

故障排除：在 PE1 上面打开调试开关 **debugging ip policy** 即可察看策略路由的转发流程。

3.5.2 OSPF 多实例故障诊断与排错

故障之一：sham link 接口不能 up

故障排除：用 **display ip route vpn-instance** 查看有没有到 sham link 目的地址的与该 OSPF 进程属于同一个 vpn-instance 的 BGP 路由。如果该路由不存在，则 sham link 不能 up。

第4章 MPLS L2VPN

4.1 MPLS L2VPN 概述

4.1.1 MPLS L2VPN 概述

基于 ATM 或 FR 的 VPN 应用非常广泛，它们能在不同 VPN 间共享运营商的网络结构。但是这种传统的 VPN 具有一些缺陷：

- 依赖于专用的媒介（如 ATM 或 FR）。要提供基于 ATM 的 VPN 服务，运营商必须建立一张覆盖全国的 ATM 网络；要提供基于 FR 的服务，又需要建立一张覆盖全国的 FR 网络。这造成了网络建设上的极大浪费；
- VPN 的部署比较复杂，特别是向现有的 VPN 加入一个站点时，需要同时修改所有接入此 VPN 站点的边缘节点的配置。

传统 VPN 的这些缺点导致了一些替代方案的产生，MPLS L2VPN 就是其中的一种。顾名思义，MPLS L2VPN 提供基于 MPLS 网络的二层 VPN 服务。从用户的角度来看，这个 MPLS 网络就是一个二层的交换网络，通过这个网络，可以在不同站点之间建立二层的连接。以 ATM 为例，每一个用户边缘设备（CE）配置一个 ATM 虚电路，通过 MPLS 网络与远端的另一个 CE 设备相连，与通过 ATM 网络实现互联是完全一样的。

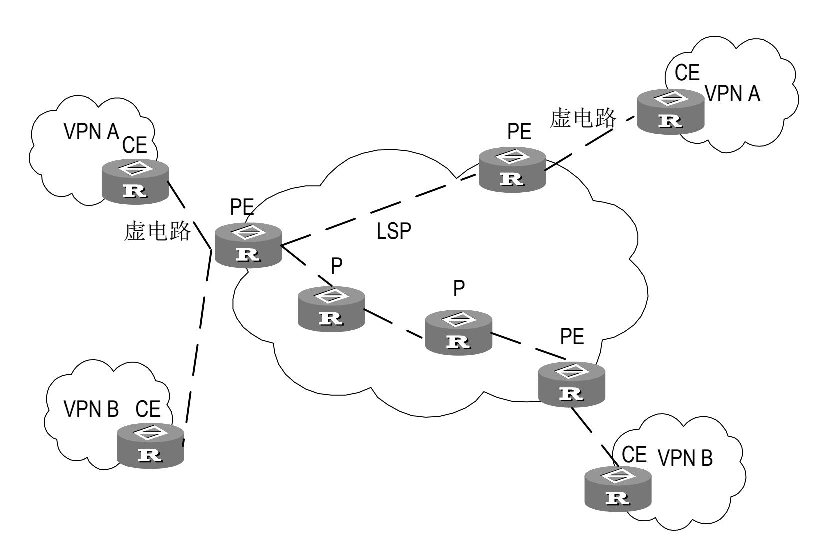


图4-1 MPLS L2VPN 组网示意图

MPLS L2VPN 具有以下优点：

- 支持多种链路层协议。可以在统一的 MPLS 网络上提供基于不同媒介的二层 VPN 服务，包括 ATM（AAL5、ATM cell relay）、FR、VLAN、Ethernet、PPP、HDLC 等。
- 支持多种网络层协议，如 IP、IPV6、IPX、SNA 等。可以提供多种服务，如三层 VPN、流量工程和 QoS 等，极大地节省网络建设的投资。
- 可伸缩性强。MPLS L2VPN 只为用户建立二层连接关系，不需引入和管理用户的路由信息。这样大大减轻了 PE 设备和整个 SP（运营商）网络的负担，从而使运营商能支持更多的 VPN 和接入更多的用户。
- 可靠性和用户路由的私有性得到保证。由于不引入用户的路由信息，MPLS L2VPN 不需要也不能获得和处理用户路由，保证了用户路由的私有性。

4.1.2 MPLS L2VPN 帧格式

MPLS L2VPN 帧格式为：

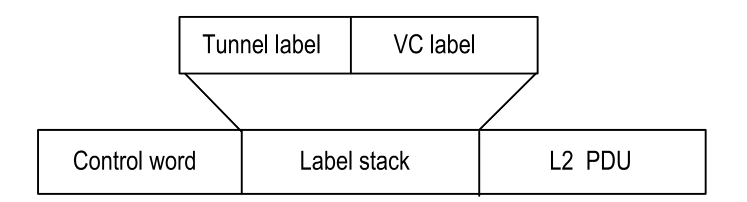


图4-2 MPLS L2VPN 帧格式

Control word：一般情况下，在 MPLS 网络上传输 l2vpn 报文时没有必要传送整个的二层帧，只需在入口处将二层帧头剥离，然后在出口端重新添加即可。但有些情况下，如链路层封装为 AAL5 及 FR 时，二层帧头中有些信息需要携带，为了解决这个问题提出了控制字的概念，利用控制字来携带必要的信息，这些信息都是 INGRESS 端和 EGRESS 端协商好的。

隧道标签（外层标签）：MPLS 标签或 GRE 标签，用于将报文从一个 PE 传递到另一个 PE。

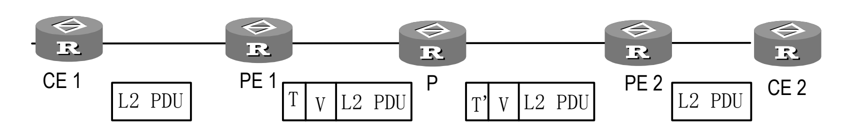
VC 标签（内层标签）：用来标识 PE-CE 间链路的底层标签，Martini 方式和 Kompella 方式的 VC 标签含义不同，CCC 方式的 MPLS L2VPN 没有这层标签。

MPLS L2VPN 支持的链路层封装类型有 ATM AAL5、ATM cell relay、FR、cisco HDLC、PPP、VLAN、ethernet 等。目前要求同一个 VPN 的各个节点使用统一的封装类型。

4.1.3 报文转发过程

在 MPLS L2VPN 中，CE、PE、P 的概念与 BGP/MPLS VPN 一样，原理也很相似，它也是利用标签栈来实现用户报文在 MPLS 网络中的透明传送的。首先 PE 之间通过手工配置或通过信令协议建立隧道。当 PE 与 CE 相连的接口 UP 时，PE 为 CE 发来的报文分配 VC 标签，然后再为报文打上隧道标签。当报文在到达远端 PE 后，远端 PE 去掉隧道标签，然后根据 VC 标签将报文发到相应的 CE。

转发过程中，报文的标签栈变化如下图所示：



L2 PDU：链路层报文
T：Tunnel标记
V：VC标记
T'：转发过程中外层标记会被替换

图4-3 MPLS L2VPN 标签栈处理

4.1.4 MPLS L2VPN 的实现方式

当前 MPLS L2VPN 还没有形成正式的标准。IETF 的 PPVPN（Provider-provisioned Virtual Private Network）工作组制订了多个框架草案，其中最主要的两种称为 Martini 草案和 Kompella 草案。到 2002 年 3 月为止，这两个草案的名称分别是：

draft-martini-l2circuit-trans-mpls-08.txt

draft-kompella-ppvpn-l2vpn-01.txt

Martini 草案定义了通过建立点到点的链路来实现 MPLS L2VPN 的方法。它以 LDP 为信令协议来传递双方的 VC 标签，因此我们称之为 Martini 方式 MPLS L2VPN。

Kompella 草案则定义了怎样在 MPLS 网络上以端到端（CE 到 CE）的方式建立 MPLS L2VPN。目前它采用 BGP 为信令协议来散发二层可达信息和 VC 标签，因此我们称之为 Kompella 方式 MPLS L2VPN。

此外，也可以不使用信令，采用静态配置 VC 标签的方式来实现 MPLS L2VPN 服务。CCC 就是静态配置 MPLS L2VPN 的一种方式。

下面分别介绍 MPLS L2VPN 的特点和实现方式。

1. CCC 方式 MPLS L2VPN

CCC 是 Circuit Cross Connect（电路交叉连接）的缩写，是通过配置静态 LSP 来实现 MPLS L2VPN 的一种方式。与普通的 MPLS L2VPN 不同的是，CCC 采用一层标签，即 Tunnel 标签，来传送用户数据，因此它对 LSP 的使用是独占性的，用户

必须逐一在每个节点上（包括 PE、P）单独为每一个 CCC 连接手工配置两条 LSP（两个方向各一条），这两条 LSP 将只能用于传递这个 CCC 连接的数据，不能用于其他 MPLS L2VPN 连接，也不能用于 BGP/MPLS VPN 或承载普通的 IP 报文。

CCC 方式的最大优点是：不需要任何信令传递二层 VPN 信息，只要能支持 MPLS 转发即可，这样在不同运营商接入的 CE 可以方便的进行互连。此外，由于 LSP 是专用的，可以提供 QoS 保证。

2. SVC（Static Virtual cucuit）方式 MPLS L2VPN

SVC 方式其实是 Martini 方式的一种静态实现，不同点在于不使用 LDP 作为传递二层 VC 和链路信息的信令，手工配置 VC 标签即可。

3. Martini 方式 MPLS L2VPN

Martini 方式 MPLS L2VPN 采用扩展的 LDP（为了在 PE 之间交换 VC 标签，Martini 草案对 LDP 进行了扩展，增加了 VC FEC 的 FEC 类型）等作为传递 VC 信息的信令。它采用 VC-TYPE+VC-ID 来识别一个 VC。VC-TYPE 表明链路层封装的类型，VC-ID 则用于唯一标志一个 VC。同一个 VC-TYPE 的所有 VC 中，其 VC-ID 必须在整个 PE 中唯一。连接两个 CE 的 PE 通过 LDP 交换 VC 标签，并通过 VC-ID 将对应的 CE 绑定起来，一个 VC 就建立起来了，这样两个 CE 就可以通过这个 VC 来传递二层数据。

Martini 方式不能提供象 CCC 方式的本地交换功能，也不象 CCC 远程连接那样一条 LSP 只能被一条远程 CCC 连接独享。Martini 方式下，服务运行商网络中的一条 LSP 可以被多条 VC 共享使用。

与 Kompella 方式相比，由于它不依赖于定时刷新机制，所以对故障的感知速度要快。

Martini 方式适合稀疏的二层连接，例如星型连接。

4. Kompella 方式 MPLS L2VPN

Kompella 方式的二层 VPN 与 RFC2547 定义的三层 BGP/MPLS VPN 很相似。象 BGP/MPLS VPN 一样，各个 PE 之间通过建立的 IBGP 会话自动发现二层 VPN 的各个节点，并传递 VPN 信息。在标签分配方面，Kompella 方式 MPLS L2VPN 采取标签块的方式，标签块的大小等于 CE range（由用户指定），以便一次为多个连接分配标签。这种方式允许用户为 VPN 分配一些额外的标签，留待以后使用，这样可以减少了 VPN 部署和扩容时的配置工作量。标签块大小确定后，系统可以通过特定算法自动计算出每条连接所需要的标签，即可通过 MPLS LSP 实现报文转发。另外，与 BGP/MPLS VPN 相同，Kompella 方式 MPLS L2VPN 也使用 vpn-target 来区分不同的 VPN，这使得 VPN 组网具备了极大的灵活性。

与 Martini 方式 MPLS L2VPN 不同，Kompella 方式 MPLS L2VPN 不是直接对 CE 与 CE 之间的连接进行操作，而是在整个 SP 网络中划分不同的 VPN，在 VPN 内部

对 CE 进行编号。要建立两个 CE 之间的连接时，只需在 PE 上设置本地 CE 和远程 CE 的 CE ID，并指定本地 CE 为这个连接分配的 Circuit ID（例如 ATM 的 VPI/VCI）。

说明：

CE 的配置比较简单，只配置相关的接口即可。这里仅对 PE 的配置进行分析。

当同时配置了 L2VPN 业务和 L3VPN 业务时，以 L2VPN 业务为准；若此时去掉 L2VPN 业务，可以继续使用 L3VPN 业务。

4.2 CCC 方式 MPLS L2VPN 配置

PE 上 CCC 配置包括：

- 配置接口
- 使能 MPLS
- 配置静态 LSP
- 使能 MPLS L2VPN
- 创建 CCC 连接

P 路由器上仅需使能 MPLS、配置双向静态 LSP 即可。

4.2.1 配置与 CE 相连的接口

这里只列出一些必要的配置，可选配置请参考《VRP3.3 操作手册》链路层协议部分。

1. 配置 PPP、HDLC、以太网接口

如果连接 CE 的接口为 PPP、HDLC 或以太网接口，则进入接口视图配置链路层协议即可。

2. 配置 ATM 接口

如果连接 CE 的接口为 ATM 接口，则进入接口视图进行如下配置。

表4-1 配置 ATM 接口

操作	命令
创建 ATM PVC/进入 PVC 模式	**pvc** [*name*] *vpi*/*vci*

对于 ATM，可以使用主接口或子接口作为 L2VPN 连接的 CE 接口。

3. 配置 FR 接口

如果连接 CE 的接口为 FR 接口，则进入接口视图进行如下配置。

表4-2 配置 FR 接口

操作	命令
配置接口封装为帧中继	**link-protocol fr** [**nonstandard** | **ietf**]
配置帧中继接口类型	**fr interface-type** { **dce** | **dte** | **nni** }
配置帧中继 LMI 协议类型	**fr lmi type** { **ansi** | **nonstandard** | **q933a** }
配置静态或动态帧中继地址映射	**fr map ip** { *protocol-address* [*ip-mask*] | *default* } *dlci* [**broadcast**] [**nonstandard** | **ietf**] **fr inarp** [**ip**] [*dlci*]
为接口分配虚电路	**fr dlci** *dlci*

对于 FR，可以使用主接口或子接口作为 L2VPN 连接的 CE 接口。

4. 配置 VLAN 子接口

如果连接 CE 的接口为 VLAN 子接口，则进入子接口视图进行如下配置。

表4-3 配置 VLAN 子接口

操作	命令
设置以太网子接口或千兆以太网子接口的封装类型以及相关联的 VLAN ID	**vlan-type dot1q vid** *vid*



注意：

L2VPN 支持的接口类型包括：以太网接口/子接口、Serial 接口和 ATM 接口/子接口

对于 VLAN，只能使用以太网子接口作为 CE 接口。如果使用以太网主接口作为 CE 接口，系统会默认为是 Ethernet 封装类型，而不是 VLAN。

在 L2VPN 中，每个子接口只能配置一个虚电路。如果有多于一个虚电路，则只有第一个有效，其他虚电路无效。

4.2.2 使能 MPLS

MPLS L2VPN 依赖于 MPLS 而存在，因此配置 MPLS L2VPN 参数之前，先要使能 MPLS。

请在系统视图下进行下列配置。

表4-4 使能 MPLS

操作	命令
配置 LSR ID	**mpls lsr-id** *X.X.X.X*
使能 MPLS	**mpls**

4.2.3 配置静态 LSP

CCC 使用静态 LSP 在 SP 网络中透传二层报文。因此，在配置 CCC 连接之前，要先在两个 PE 和中间的所有 P 路由器上配置两条静态 LSP（双向）。

请在 MPLS 视图下进行下列配置。

表4-5 配置静态 LSP

操作	命令
创建静态 LSP 的 egress	**static-lsp egress** *lsp-name* **l2vpn incoming-interface** *interfac-type interfac-num* **in-label** *in-label*
删除静态 LSP 的 egress	**undo static-lsp egress** *lsp-name* **l2vpn**
创建静态 LSP 的 ingress	**static-lsp ingress** *lsp-name* { **l2vpn | destination** *ip-add*} { **nexthop** *next-hop-addr* | **outgoing-interface** *interfac-type interfac-num* } **out-label** *out-label*
删除静态 LSP 的 ingress	**undo static-lsp ingress** *lsp-name* **l2vpn**
创建静态 LSP 的 transit	**static-lsp transit** *lsp-name* **l2vpn incoming-interface** *interface-type interface-num* } **in-label** *in-label* { **nexthop** *next-hop-addr* | **outgoing-interface** *interface-type interface-num* } **out-label** *out-label*
删除静态 LSP 的 transit	**undo static-lsp transit** *lsp-name* **l2vpn**

4.2.4 使能 MPLS L2VPN

请在系统视图下进行下列配置。

表4-6 使能 MPLS L2VPN

操作	命令
使能 MPLS L2VPN 功能	**mpls l2vpn**

4.2.5 创建 CCC 连接

CCC 连接有两种：本地连接和远程连接。本地连接是在两个本地 CE 之间建立的连接。本地连接可以直接在 PE 上完成交换，不需要配置静态 LSP。远程连接是指本

地 CE 和远程 CE 之间的连接，也就是说，两个 CE 连在不同的 PE 上。这就需要配置两条两个方向的静态 LSP 以便在 PE 之间传递报文。

请在系统视图下进行下列配置。

表4-7 创建 CCC 连接

操作	命令
创建 CCC 本地连接	**ccc** *ccc-connection-name* **interface** *type number* **out-interface** *outinterface-type outinterface-num*
创建 CCC 远程连接	**ccc** *ccc-connection-name* **interface** *type number* **transmit-lsp** *transmit-lsp-name* **receive-lsp** *receive-lsp-name*

 注意：

CCC 远程连接对 LSP 的使用是独占性的：
必须为每个 CCC 远程连接创建两个静态 LSP，不能让两个 CCC 连接共用同一个静态 LSP;
静态 LSP 一旦被 CCC 远程连接使用，就不能再用于别的用途（如承载 IP 数据、承载 BGP/MPLS VPN 数据等）。因此，在为 CCC 连接创建静态 lsp 时，其 FEC 最好选择本网络中不存在的 IP 地址，以免这个 lsp 被路由选中。

4.3 SVC 方式 MPLS L2VPN 配置

4.3.1 SVC 方式 MPLS L2VPN 配置任务

PE 上 SVC 方式 L2VPN 的配置任务包括：

- 配置与 CE 相连的接口
- 使能 MPLS、MPLS LDP
- 使能 MPLS L2VPN
- 配置两个 PE 之间的隧道
- 创建 SVC 方式 MPLS L2VPN 连接

前三步的配置请参考 4.2 CCC方式MPLS L2VPN配置。

4.3.2 配置 PE 之间的隧道

当前隧道管理提供两种类型的隧道，即 GRE 和 LSP。

1. 配置 GRE 隧道

表4-8 配置 GRE 隧道

操作	命令
创建隧道接口	**Interface tunnel** num
配置隧道源地址（隧道接口模式）	**source** X.X.X.X
配置隧道目的地地址（隧道接口模式）	**destination** X.X.X.X

2. 配置 LSP 类型隧道

表4-9 配置 LSP 类型隧道

操作	命令
配置创建 LSP 的策略	**lsp-trigger** { **all** | **ip-prefix** *ip-prefix* }



注意：

因为 LSP 是 MPLS 根据路由来创建的，所以为了创建到达目的地的 LSP，应先保证各 PE 的主机路由要通过路由协议传播到对端。

4.3.3 创建 SVC 方式 MPLS L2VPN 连接

请在接口视图下进行下列配置。

表4-10 创建 SVC 方式 MPLS L2VPN 连接

操作	命令
创建 SVC 方式 MPLS L2VPN 连接	**mpls static-l2vc destination** *destination-router-id* **transmit-vpn-label** *transmit-label-value* **receive-vpn-label** *receive-label-value*
删除 SVC 方式 MPLS L2VPN 连接	**undo mpls static-l2vc**

此命令应该配在 PE 的私网接口上，即连接 CE 的接口。



注意：

SVC 方式 L2VPN 的收、发标签的合法性由用户保证。

4.4 Martini 方式 MPLS L2VPN 配置

PE 上 Martini 方式 MPLS L2VPN 的配置包括：

- 配置与 CE 相连的接口
- 使能 MPLS、MPLS L2VPN
- 使能 MPLS LDP，并配置 LDP Remote Peer
- 配置 PE 之间的隧道
- 创建 Martini 方式 MPLS L2VPN 连接

前两步的配置方法与CCC完全相同，请参考 4.2 CCC方式MPLS L2VPN配置。配置PE之间的隧道请参考 4.3 SVC方式MPLS L2VPN配置。

4.4.1 配置 LDP Remote Peer

Martini 方式的 MPLS L2VPN 依赖于 LDP 的 Remote Peer 来交换 VC 标签，因此在配置连接之前，要先在沿途各个路由器上使能 LDP，并配置 LDP 的 Remote Peer。

LDP 的 Remote Peer 的相关配置请参考 MPLS 模块 LDP 配置。

4.4.2 创建 Martini 方式 MPLS L2VPN 连接

在 PE 上配置 Martini 方式 MPLS L2VPN 连接需要指定对端 PE 的 IP 地址（lsr-id）及 VC ID，其中 VC ID 与封装类型的组合必须在 PE 上唯一。

请在接口视图下进行下列配置。

表4-11 创建 Martini 方式的 MPLS L2VPN 连接

操作	命令
创建 Martini 方式的 MPLS L2VPN 虚连接	**mpls l2vc** *ip-address vc-id*
删除 Martini 方式的 MPLS L2VPN 虚连接	**undo mpls l2vc**

此命令应该配在 PE 的私网接口上，即连接 CE 的接口。



注意：

Martini 方式 MPLS L2VPN 要求在一台 PE 设备上，同一种封装类型下的 VC ID 必须唯一，而改封装有可能会造成 VC ID 的冲突。例如，serial0 和 serial1 接口分别封装 HDLC 和 PPP，各创建了一个 LDP 方式连接，VC ID 都是 1。如果将 serial1 的链路层封装类型改为 HDLC，则会造成两个 CCC-HDLC 封装的 LDP 方式连接，其 VC ID 都是 1。为了避免 VC ID 冲突的情况，此时 serial1 上的 LDP 方式连接会被自动删除。

4.5 Kompella 方式 MPLS L2VPN 配置

Kompella 方式 MPLS L2VPN 的配置包括：

- 配置与 CE 相连的接口
- 使能 MPLS 以及 MPLS L2VPN
- 配置 BGP 参数
- 创建和配置 VPN
- 创建 CE 并配置 CE 的连接

前两步的配置方法与CCC完全相同，但为今后扩容需要应预先配置一些预留的接口，配置方法请参考 4.2 CCC方式MPLS L2VPN配置。

4.5.1 配置 BGP 参数

Kompella 方式 MPLS L2VPN 以扩展 BGP 为信令协议来分发 VC 标签，所以还需要在 PE 上配置 BGP 参数。关于 BGP 参数的配置请参见路由协议模块的 BGP 配置部分，MPLS L2VPN 本身对 BGP 的配置没有特殊要求。

完成了 BGP 相关配置，需要在 MPLS L2VPN 地址族视图下激活对等体组的参数，具体命令如下：

1. 进入 L2VPN 地址族视图

请在 BGP 视图下进行下列配置。

表4-12 进入 L2VPN 地址族视图

操作	命令
进入 MPLS L2VPN 地址族视图	**l2vpn-family**
退出 MPLS L2VPN 地址族视图	**undo l2vpn-family**

执行 **undo** 命令后，将退回到 BGP 视图，并删除 L2VPN 地址族视图。

2. 激活对等体（组）

请在 L2VPN 地址族视图下进行下列配置。

表4-13 激活对等体/对等体组

操作	命令
激活指定对等体（组）	**peer** { *peer-address* | *group-name* } **enable**
去激活指定对等体（组）	**undo peer** { *peer-address* | *group-name* } **enable**

缺省情况下，只有 BGP 的 IPv4 单播地址族的对等体才是激活的，其它类型的对等体（组）都是去激活的，不能交换路由信息。

4.5.2 创建和配置 VPN

1. 创建 VPN

对于 BGP 方式的 MPLS L2VPN，必须在 PE 上创建 MPLS L2VPN VPN，同时指定封装类型，其中封装类型应与 CE 接口的封装类型一致。

请在系统视图下配置下面命令。

表4-14 创建 VPN

操作	命令
创建 VPN，并指定封装方式	**mpls l2vpn** *vpn-name* [**encapsulation** { **atm-aal5** | **ethernet** | **fr** | **hdlc** | **ppp** | **vlan**}]
进入已经配置的 MPLS L2VPN 视图	**mpls l2vpn** *vpn-name*
删除已经配置的 MPLS L2VPN	**undo mpls l2vpn** *vpn-name*

2. 配置 VPN

在 MPLS L2VPN 的 VPN 视图下配置 vpn target 和 RD。vpn target 和 RD 的意义和用法与 BGP/MPLS VPN 完全相同，这里不再赘述。说明一点：对于一个 MPLS L2VPN，必须先配置 RD，然后才能配置其他命令。而且 RD 配置后不能修改。修改 RD 的唯一办法是：删除这个 MPLS L2VPN，然后，重新创建。

此外，用户还可以为 VPN 配置二层 MTU。VPN 的 mtu 应该在全网统一。如果在两个 PE 上同一个 VPN 的 mtu 不同，则这两个 PE 之间无法正常交换可达信息，也无法建立连接。

请在 MPLS L2VPN 视图下配置下面命令。

表4-15 配置 VPN

操作	命令
配置 MPLS L2VPN 的 RD	**route-distinguisher** *route-distinguisher*
配置 MPLS L2VPN 的 vpn-target	**vpn-target** vpn-target-ext-community [**import-extcommunity**| **export-extcommunity** | **both**]

删除 MPLS L2VPN 的 vpn-target	**undo vpn-target** *vpn-target-ext-community* [**import-extcommunity** | **export-extcommunity** | **both**]
配置 VPN 的二层 mtu	**mtu** *mtu*

4.5.3 创建 CE 并配置 CE 的连接

1. 创建 CE

在 PE 上创建的 CE 应与实际相连的 CE 设备一一对应，需要用户为 CE 指定一个唯一的 ID，也可以指定 CE range 的大小。

请在 MPLS L2VPN 视图下进行下列配置。

表4-16 创建 CE

操作	命令
创建 CE/修改 CE range	**ce** *name* **id** *id* [**range** *range*] [**default-offset** *offset*]
进入已经配置的 CE	**ce** *name*
删除 CE	**undo ce** *name*

CE ID 用于在一个 VPN 中唯一确定一个 CE，必须在整个 VPN 中唯一。为了方便配置，CE ID 最好从 1 开始，采用连续自然数编号。

CE range 表明这个 CE 最多能与多少个 CE 建立连接。在标签资源足够丰富的情况下（一般来说，标签资源总是足够丰富的），可以根据对这个 VPN 规模发展的预计，把 CE range 设置得比实际需要的大一点。这样当以后对 VPN 进行扩容，增加 VPN 中的 CE 数目时，就可以尽量少的修改配置。

如果对 VPN 扩容时，发现原来设置的 CE range 比所需要的更小。例如：扩容后需要连接的 CE 数目为 20，但 CE range 为 10。此时可以把 CE range 修改为 20。修改 CE range 时，为了保证原来的 10 个连接不会中断，系统不是把原来的标签块释放，重新申请大小为 20 的标签块，而是在原来的标签块之外，重新申请一个新的标签块，大小为 10。所以修改 CE range 不会导致原来业务的中断。

假设一个企业的 VPN 包括 10 个 CE，但是考虑到企业会扩展业务，将来可能会有 20 个 CE。这样可以把每个 CE 的 CE range 设置为 20，系统会预先为未来的 10 个 CE 分配标签。以后 VPN 添加 CE 节点时，配置的修改仅限于与新 CE 直接相连的 PE，其他 PE 不需要作任何修改。这使得 VPN 的扩容变得非常简单。



注意：

修改 CE range 只能把 CE range 变大，不能变小。例如：原来的 CE range 为 10，则可以把它改为 20，但如果想改为 5，则会失败。把 CE range 改小的唯一方法是：删除这个 CE，重新创建。

2. 配置 CE 的连接

在 MPLS L2VPN CE 视图下，可以创建 CE 的连接。创建连接时，指定连接的 CE 接口和对端 CE 的编号（即 CE offset）即可。为了今后扩容的需要，也可以预先配置一些预留的连接。

在 MPLS L2VPN CE 视图下配置下面命令。

表4-17 创建 CE

操作	命令
创建连接	**connection** [**ce-offset** *offset*] { **interface** *interface-type interface-num* }
删除连接	**undo connection** [**ce-offset** *offset*] { **interface** *interface-type interface-num* }

创建连接时，CE offset 是一个可选参数。如果用户没有指定 CE offset，有两种情况：

(1) 如果这是这个 CE 的第一个连接，则 CE offset 默认为 1；

(2) 否则，CE offset 是上一个连接的 CE offset+1。

所以，如果在规划 VPN 时，将 CE ID 编号从 1 顺序递增；然后在配置连接时按 CE ID 顺序配置，则大多数连接都可以省略 CE offset 参数，使用默认的 CE offset，从而简化配置。

4.6 MPLS L2VPN 显示与调试

在完成上述配置后，在所有视图下执行 **display** 命令可以显示配置后 MPLS L2VPN 的运行情况，通过查看显示信息验证配置的效果。

在用户视图下，执行 **debugging** 命令可对 MPLS L2VPN 进行调试。

表4-18 L2VPN 监控维护命令

操作	命令
显示 CCC 连接信息	**display ccc** [*ccc-name* | **type** [**local** | **remote**]]
显示 SVC 方式 L2VPN 连接诶信息	**display mpls static-l2vc** [**interface** *interface-type interface -num*]

操作	命令
显示Martini方式MPLS L2VPN连接信息	**display mpls l2vc** [**interface** *interface-type interface-num* | **verbose**]
显示指定接口下的 L2VPN 信息	**display mpls l2vpn forwarding-info** [*vc-label*] **interface** *interface-type interface-num*
显示系统运行信息，可显示所有L2VPN 信息。	**display bgp l2vpn all**
打开 Kompella 方式 MPLS L2VPN 调试信息开关	**debugging bgp** [{ { **keepalive** | **open** | **packet** | **update** | **route-refresh** } [**receive** | **send**] [**verbose**] | **event** }]
打开 MPLS L2VPN 调试信息开关	**debugging mpls l2vpn { all | advertisement | error | event | connections [interface *interface-type interface-num*] }**

4.7 MPLS L2VPN 典型配置举例

4.7.1 CCC 方式 MPLS L2VPN 配置举例

1. 组网需求

CE 与 PE 之间通过 serial 口相连，链路层封装 PPP。要求在 CE-A 和 CE-B 之间建立一个本地连接，CE-A 和 CE-C 之间建立一个远程连接。

2. 组网图

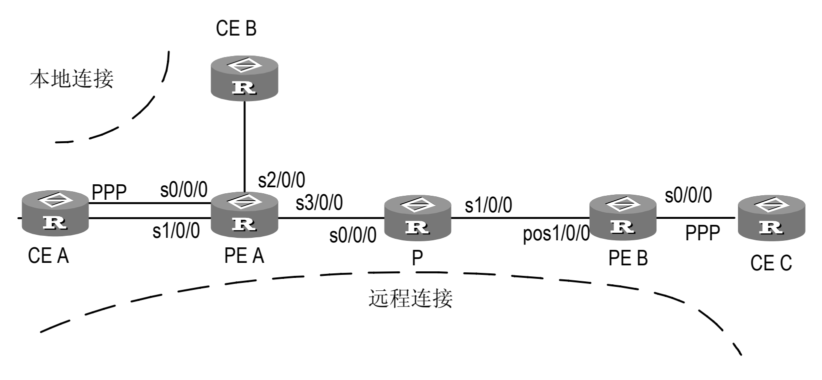


图4-4 CCC 组网图

3. 配置步骤

(1) 配置 PE-A

全局使能 MPLS。

```
[Quidway] mpls lsr-id 172.1.1.1
[Quidway] mpls
```

全局使能 MPLS L2VPN。

```
[Quidway] mpls l2vpn
```

配置接口 SERIAL 0/0/0。

```
[Quidway] interface serial0/0/0
[Quidway-Serial0/0/0] link-protocol ppp
```

配置接口 SERIAL 1/0/0。

```
[Quidway] interface serial 1/0/0
[Quidway-Serial1/0/0] link-protocol ppp
```

配置接口 SERIAL2/0/0。

```
[Quidway] interface serial 2/0/0
[Quidway-Serial2/0/0] link-protocol ppp
```

在接口 SERIAL 3/0/0 上使能 MPLS。

```
[Quidway] interface serial 3/0/0
[Quidway-Serial3/0/0] link-protocol ppp
[Quidway-Serial3/0/0] mpls
```

配置本地连接。

```
[Quidway] ccc local-conn interface serial0/0/0 outgoing-interface serial2/0/0
```

配置静态 LSP，出标签 100，出接口 SERIAL 3/0/0。

```
[Quidway] mpls
[Quidway-mpls] static-lsp ingress PEA-PEB destination 10.0.0.0 l2vpn
outgoing-interface serial3/0/0 out-label 100 pre 2 metric 3
```

配置静态 LSP，入标签 201，入接口 SERIAL 3/0/0。

```
[Quidway-mpls] static-lsp egress PEB-PEA l2vpn incoming-interface serial3/0/0
in-label 201
```

配置远程连接

```
[Quidway] ccc remote-connection interface serial3/0/0 transmit-lsp PEA-PEB
receive-lsp PEB-PEA
```

(2) 配置 PE-B

全局使能 MPLS。

```
[Quidway] mpls lsr-id 10.0.0.1
[Quidway] mpls
```

全局使能 MPLS L2VPN。

```
[Quidway] mpls l2vpn
```

配置接口 SERIAL 0/0/0。

```
[Quidway] interface serial 0/0/0
```

```
[Quidway-Serial0/0/0] link-protocol ppp
```

在接口 SERIAL1/0/0 上使能 MPLS。

```
[Quidway] interface serial 1/0/0
[Quidway-Serial1/0/0] link-protocol ppp
[Quidway-Serial1/0/0] mpls
```

配置静态 LSP，出标签 100，出接口 SERIAL 1/0/0。

```
[Quidway-mpls] static-lsp ingress PEB-PEA destination 10.0.0.0 l2vpn outgoing-interface serial 1/0/0 out-label 200 pre 2 metric 3
```

配置静态 LSP，入标签 201，入接口 SERIAL 1/0/0。

```
[Quidway-mpls] static-lsp egress PEA-PEB l2vpn incoming-interface serial 1/0/0 in-label 101
```

配置远程连接

```
[Quidway] ccc remote-conn interface serial1/0/0 transmit-lsp PEB-PEA receive-lsp PEA-PEB
```

(3) 配置 P

```
[Quidway] mpls lsr-id 10.0.0.2
[Quidway] mpls
[Quidway] mpls l2vpn
```

配置静态 LSP，入标签 100，入接口 SERIAL0/0/0，出标签 101，出接口 SERIAL1/0/0。

```
[Quidway-mpls] static-lsp transit PEA-PEB l2vpn incoming-intergace serial 0/0/0 in-label 100 out1going-interface serial 1/0/0 out-label 101
```

配置静态 LSP，入标签 200，入接口 SERIAL 1/0/0，出标签 201，出接口 SERIAL 0/0/0。

```
[Quidway-mpls] static-lsp transit PEB-PEA l2vpn incoming-interface serial1/0/0 in-label 200 outgoing-interface serial 0/0/0 out-label 201
```

 注意：

- CCC 本地连接 UP 的条件是:

两个 CE 接口的物理状态是 UP 的。

两个 CE 接口的封装类型一致并且是当前 MPLS L2VPN 可以支持的封装类型。

- 若对于 MPLS L2VPN 为 VLAN 封装的二层连接，两端的 CE 接口的 VLAN ID 可以一致，也可以不一致。

4.7.2 SVC 方式 MPLS L2VPN 配置举例

1. 组网需求

配置从 PE1 的 CE1 到 PE3 的 CE4 之间的一条 frame relay 封装类型的远程 SVC 连接。

📖 **说明：**

为了 CE1 ~ CE4 正常通信，其 DCE/DTE 的设置要匹配。在这里的 CE1－PE1－PE3－CE4 的路径上，设定 CE1 为 DCE、PE1 为 DTE、PE3 为 DCE、CE4 为 DTE。

2. 组网图

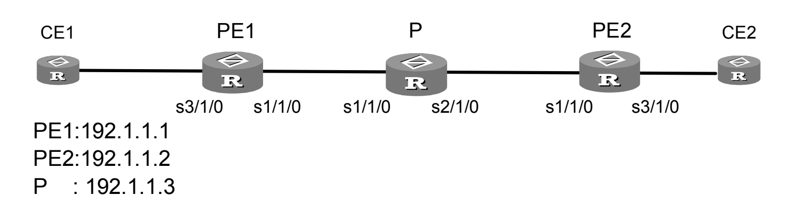


图4-5 SVC 方式 MPLS L2VPN 配置举例

3. 配置步骤

(1) 配置 PE1

配置 LSR ID，使能 MPLS、LDP 及 MPLS L2VPN。

```
[Quidway-mpls] mpls lsr-id 192.1.1.1
[Quidway] mpls
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

配置 CE 接口。

```
[PE1] interface serial3/1/0
[PE1-serial3/1/0] fr dlci 101
```

配置 serial 口。

```
[PE1] interface serial1/1/0
[PE1-serial1/1/0] mpls
[PE1-serial1/1/0] mpls ldp enable
[PE1-serial1/1/0] ip address 168.1.1.1 255.255.0.0
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway -ospf-1] area 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 192.1.1.1 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立

```
[PE1] mpls
[PE1-mpls] lsp-trigger all
```

配置 SVC 方式连接

```
[PE1] interface serial3/1/0
[PE1-s3/1/0] mpls static-l2vc destination 192.1.1.3 transmit-vpn-label 111
receive-vpn-label 333
```

(2) 配置 PE2

配置 LSR ID，使能 MPLS、LDP 及 MPLS L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.2
[Quidway] mpls
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

配置 CE 接口。

```
[PE2] interface serial3/1/0
[PE2-serial3/1/0] fr linterface-type dce
[PE2-serial3/1/0] fr dlci 101
```

配置 serial 口。

```
[PE1] interface serial1/1/0
[PE1-serial1/1/0] mpls
[PE1-serial1/1/0] mpls ldp enable
[PE1-serial1/1/0] ip address 169.1.1.1 255.255.0.0
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway-ospf-1] area 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 192.1.1.2 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 169.1.0.0 0.0.255.255
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立

```
[PE2] mpls
[PE2-mpls] lsp-trigger all
```

配置 SVC 方式连接。

```
[PE2-s3/1/0] mpls static-l2vc destination 192.1.1.1 transmit-vpn-label 333
receive-vpn-label 111
```

(3) 配置 P

配置 LSR ID，使能 MPLS、LDP 及 MPLS L2VPN。

```
[Quidway-mpls] mpls lsr-id 192.1.1.2
[Quidway] mpls
```

```
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

配置 serial 口。

```
[PE2] interface serial1/1/0
[PE2- serial1/1/0] ip address 168.1.1.2 255.255.0.0
[PE2] interface serial2/1/0
[PE2- serial2/1/0] ip address 169.1.1.2 255.255.0.0
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立

```
[PE2] mpls
[PE2-mpls] lsp-trigger all
```

4.7.3 Martini 方式 MPLS L2VPN 配置举例

1. 组网需求

CE 采用 VLAN 接入，CE-A 和 CE-B 之间建立一个远程连接。

2. 组网图

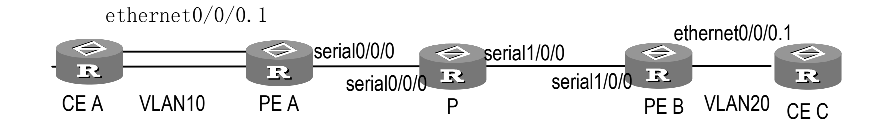


图4-6 Martini 方式 MPLS L2VPN 组网图

3. 配置步骤

(1) 配置 PE-A

配置 LSR ID，使能 MPLS、LDP 及 MPLS L2VPN。

```
[Quidway-mpls] mpls lsr-id 192.1.1.1
[Quidway] mpls
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

配置 VLAN 子接口。

```
[Quidway] interface ethermet0/0/0.1
[Quidway-Ethernet0/0/0.1] vlan-type dot1q vid 20
```

配置 serial 口。

```
[Quidway] interface serial 0/0/0
[Quidway-serial0/0/0] ip address 168.1.1.1 255.255.0.0
[Quidway-serial 0/0/0] mpls
[Quidway-serial 0/0/0] mpls ldp enable
```

配置 loopback 接口地址，作为 route id 使用。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.1 255.255.255.255
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway-ospf-1] area 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 192.1.1.1 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 168.1.0.1 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 192.2.1.0 0.0.0.255
```

#创建 GRE 隧道。

```
[Quidway] interface tunnel 1
[Quidway-Tunnel1] ip address 192.2.1.1 255.255.255.0
[Quidway-Tunnel1] source 192.1.1.2
[Quidway-Tunnel1] destination 192.1.1.1
```

配置 ldp remote peer。

```
[Quidway] mpls ldp remote-peer 1
[Quidway-remote-peer-1] remote-peer 192.1.1.2 255.255.255.255
```

配置 martini 方式的 MPLS L2VPN 连接。

```
[Quidway] interface Ethernet0/0/0.1
[Quidway-Ethernet0/0/0.1] mpls l2vc 192.1.1.2 20
```

(2) 配置 PE-B

配置 LSR ID，使能 MPLS、LDP 以及 MPLS L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.2
[Quidway] mpls
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

配置 VLAN 子接口。

```
[Quidway-LoopBack0] interface ethernet0/0/0.1
[Quidway-Ethernet0/0/0.1] vlan-type dot1q vid 20
```

配置 Serial 口。

```
[Quidway] interface serial 1/0/0
[Quidway-serial1/0/0] ip address 169.1.1.1 255.255.0.0
[Quidway-serial1/0/0] mpls
[Quidway-serial1/0/0] mpls ldp enable
```

为 loopback 接口配置一个地址，作为 LSR ID 使用。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.2 255.255.255.255
```

启动 OSPF

```
[Quidway] ospf 1
[Quidway-ospf-1] area 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 192.1.1.2 0.0.0.0
[Quidway-ospf-1-area-0.0.0.0] network 169.1.0.0 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 192.2.0.0 0.0.0.255
```

创建 GRE 隧道

```
[L2VPN]interface tunnel 1
[L2VPN-Tunnel1] ip address 192.2.1.2 255.255.255.0
[L2VPN-Tunnel1] source 192.1.1.1
[L2VPN-Tunnel1] destination 192.1.1.2
```

配置 ldp remote peer。

```
[Quidway] mpls ldp remote-peer 1
[Quidway-mpls-remote1] remote-peer 192.1.1.1 255.255.255.255
```

配置 martini 方式的 MPLS L2VPN 连接。

```
[Quidway] interface Ethernet0/0/0.1
[Quidway-Ethernet0/0/0.1] mpls l2vc 192.1.1.1 20
```

(3) 配置 P

配置 LSR ID，使能 MPLS、LDP 及 L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.3
[Quidway] mpls
[Quidway] mpls ldp
[Quidway] mpls l2vpn
```

为 loopback 接口配置一个地址，作为 LSR ID 使用。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.3 255.255.255.255
```

配置 Serial 口。

```
[Quidway-LoopBack0] interface serial0/0/0
[Quidway-serial0/0/0] mpls
[Quidway-serial0/0/0] mpls ldp enable
[Quidway-serial0/0/0] ip address 168.1.1.2 255.255.0.0
[Quidway] interface serial1/0/0
[Quidway-serial0/0/0] mpls
[Quidway-serial0/0/0] mpls ldp enable
[Quidway-serial1/0/0] ip address 169.1.1.2 255.255.0.0
```

启动 OSPF

```
[Quidway] ospf 1
[Quidway-ospf-1] area 0.0.0.0
```

```
[Quidway-ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 169.1.0.0 0.0.255.255
[Quidway-ospf-1-area-0.0.0.0] network 192.1.1.3 0.0.0.0
```

⚠ 注意：

一个 LDP 方式二层 VPN 连接 UP 的条件有：

- 两端的 CE 接口的物理状态 UP;
- 两个 PE 间存在两条相反方向的隧道，这隧道可以是 GRE 隧道，也可以是 LSP 隧道;
- 两端 CE 接口的封装类型相同，并且为 L2VPN 支持的封装类型;
- PE 之间存在 LDP remote session，并且其状态为 UP 的。

为建立隧道，不管是 GRE 还是 LSP 类型，都需要存在到对端 PE 的路由，所以需要在路径上的各个路由器上配置 IGP，如 OSPF。

4.7.4 Kompella 方式 MPLS L2VPN 的典型配置举例

1. 配置需求

CE 与 PE 之间通过 ATM 接口相连。要求创建一个全连接的 VPN，VPN 中包含 CE-A、CE-B、CE-C 三个 CE。考虑以后可能再扩展两个 CE，每个 CE 的 CE range 设置为 4，并且为以后可能增加的两个 CE 预先分配虚电路。这样以后添加 CE 时，只需对新增 CE 的 PE 进行配置，而不需要修改其他 PE 的配置。

2. 组网图

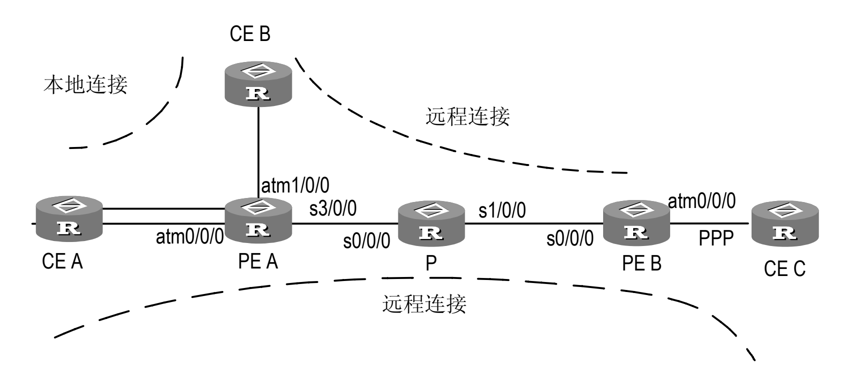


图4-7 Kompella 方式 MPLS L2VPN 组网图

3. 配置步骤

(1) 配置 PE-A （PE-C 的配置与 PE-A 相似，不再赘述）

配置 MPLS 的 LSR ID，使能 MPLS L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.1
[Quidway] mpls
```

```
[Quidway] mpls l2vpn
```

为 loopback 接口配置一个地址。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.1 255.255.255.255
```

配置 serial 口。

```
[Quidway-LoopBack0] interface serial 3/0/0
[Quidway-Serial3/0/0] ip address 168.1.1.1 255.255.0.0
[Quidway-Serial3/0/0] mpls ldp enable
```

配置 CE A 的 CE 接口。

```
[Quidway] interface atm0/0/0.1
[Quidway-Atm0/0/0.1] pvc CEA-CEB 100/101
[Quidway-Atm0/0/0.1] interface atm0/0/0.2
[Quidway-Atm0/0/0.2] pvc CEA-CEC 100/102
```

为以后扩展的 CE 预留 PVC。

```
[Quidway-Atm0/0/0.1] interface atm0/0/0.3
[Quidway-Atm0/0/0.3] pvc CEA-CED 100/103
[Quidway-Atm0/0/0.3] interface atm0/0/0.4
[Quidway-Atm0/0/0.4] pvc CEA-CEE 100/104
```

配置 CE-B 的 CE 接口。

```
[Quidway] interface atm1/0/0.1
[Quidway-Atm1/0/0.1] pvc CEB-CEA 200/101
[Quidway] interface atm1/0/0.2
[Quidway-Atm1/0/0.2] pvc CEB-CEC 200/102
```

为以后扩展的 CE 预留 PVC。

```
[Quidway] interface atm1/0/0.3
[Quidway-Atm1/0/0.3] pvc CEB-CED 200/103
[Quidway] interface atm1/0/0.4
[Quidway-Atm1/0/0.4] pvc CEB-CEE 200/104
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway -ospf-1] area 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 192.1.1.1 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立。

```
[PE1] mpls
[PE1-mpls] lsp-trigger all
```

配置 BGP 参数。

```
[Quidway] bgp 100
[Quidway-bgp] group 192 internal
[Quidway -bgp] peer 192.1.1.2 connect-interface LoopBack0
[Quidway -bgp] peer 192.1.1.2 group 192 as-number 100
[Quidway -bgp] peer 192.1.1.3 connect-interface LoopBack0
[Quidway -bgp] peer 192.1.1.3 group 192 as-number 100
[Quidway -bgp] l2vpn-family
[Quidway -bgp-af-l2vpn] peer 192 enable
```

配置 Kompella 方式 MPLS L2VPN。

```
[Quidway] mpls l2vpn vpna encapsulation atm-aal5
[Quidway-l2vpn-vpna] route-distinguisher 100:1
[Quidway-l2vpn-vpna] vpn-target 100:1 both
```

创建 CE-A，并配置本地连接、远程连接及预留连接。

```
[Quidway-l2vpn-vpna] ce ce-a id 1 range 4
```

与 CE-B 的本地连接。

```
[Quidway-l2vpn-vpna-ce-ce-a] connection ce-offset 2 atm0/0/0.1
```

与 CE-C 的远程连接。

```
[Quidway-l2vpn-vpna-ce-ce-a] connection atm0/0/0.2
```

为以后扩展预留的连接。

```
[Quidway-l2vpn-vpna-ce-ce-a] connection atm0/0/0.3
[Quidway-l2vpn-vpna-ce-ce-a] connection atm0/0/0.4
```

创建 CE-B。

```
[Quidway-l2vpn-vpna] ce ce-b id 2 range 4
```

与 CE-A 的本地连接。

```
[Quidway-l2vpn-vpna-ce-ce-b] connection ce-offset 3 atm1/0/0.1
```

与 CE-C 的远程连接。

```
[Quidway-l2vpn-vpna-ce-ce-b] connection atm1/0/0.2
```

为以后扩展预留的连接。

```
[Quidway-l2vpn-vpna-ce-ce-b] connection atm1/0/0.3
[Quidway-l2vpn-vpna-ce-ce-b] connection atm1/0/0.4
```

(2) 配置 PE-B

配置 MPLS RD，全局使能 MPLS，并全局使能 MPLS L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.2
[Quidway] mpls
[Quidway] mpls l2vpn
```

为 loopback 接口配置一个地址。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.2 255.255.255.255
```

配置 serial 口。

```
[Quidway-LoopBack0] interface serial 0/0/0
[Quidway-Serial0/0/0] ip address 169.1.1.1 255.255.0.0
[Quidway-Serial0/0/0] mpls ldp enable
```

配置 ATM 子接口和 PVC。

```
[Quidway] interface atm0/0/0.1
[Quidway-Atm0/0/0.1] pvc CEC-CEA 300/101
[Quidway] interface atm0/0/0.2
[Quidway-Atm0/0/0.2] pvc CEC-CEB 300/102
```

为以后扩展的 CE 预留 PVC。

```
[Quidway-Atm0/0/0.2] interface atm0/0/0.3
[Quidway-Atm0/0/0.3] pvc CEC-CED 300/103
[Quidway] interface atm0/0/0.4
[Quidway-Atm0/0/0.4] pvc CEC-CEE 300/104
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway -ospf-1] area 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 192.1.1.1 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立。

```
[PE1] mpls
[PE1-mpls] lsp-trigger all
```

配置 BGP 参数。

```
[Quidway] bgp 100
[Quidway-bgp]
[Quidway -bgp] peer 192.1.1.1 connect-interface LoopBack0
[Quidway -bgp] peer 192.1.1.1 group 192 as-number 100
[Quidway -bgp] peer 192.1.1.3 connect-interface LoopBack0
[Quidway -bgp] peer 192.1.1.3 group 192 as-number 100
[Quidway -bgp] l2vpn-family
[Quidway -bgp-af-l2vpn] peer 192 enable
```

配置 Kompella 方式 MPLS L2VPN。

```
[Quidway] mpls l2vpn vpna encasulation atmaal5
[Quidway-l2vpn-vpna] route-distinguish 100:1
[Quidway-l2vpn-vpna] vpn-target 100:1 both
```

创建 CE-C。

```
[Quidway-l2vpn-vpna-ce-ce-c] ce ce-c id 3 range 4
```

与 CE-A 的连接。

```
[Quidway-l2vpn-vpna-ce-ce-c] connection atm0/0/0.1
```

与 CE-B 的连接。

```
[Quidway-l2vpn-vpna-ce-ce-c] connection atm0/0/0.2
```

为以后扩展预留的连接。

```
[Quidway-l2vpn-vpna-ce-ce-c] connection atm0/0/0.3 ce-offset 4
[Quidway-l2vpn-vpna-ce-ce-c] connection atm0/0/0.4
```

(3) 配置 P

配置 MPLS RD，全局使能 MPLS，并全局使能 MPLS L2VPN。

```
[Quidway] mpls lsr-id 192.1.1.3
[Quidway] mpls
[Quidway] mpls ldp
```

为 loopback 接口配置一个地址。

```
[Quidway] interface loopback 0
[Quidway-LoopBack0] ip address 192.1.1.3 255.255.255.255
```

配置串口。

```
[Quidway-LoopBack0] interface serial 0/0/0
[Quidway-Serial0/0/0] link-protocol ppp
[Quidway-Serial0/0/0] ip address 168.1.1.2 255.255.0.0
[Quidway-Serial0/0/0] mpls
[Quidway-Serial0/0/0] mpls ldp enable
[Quidway] interface serial 1/0/0
[Quidway-Serial1/0/0] link-protocol ppp
[Quidway-Serial1/0/0] ip address 169.1.1.2 255.255.0.0
[Quidway-Serial1/0/0] mpls
[Quidway-Serial1/0/0] mpls ldp enable
```

启动 OSPF。

```
[Quidway] ospf 1
[Quidway -ospf-1] area 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 192.1.1.3 0.0.0.0
[Quidway -ospf-1-area-0.0.0.0] network 168.1.0.0 0.0.255.255
[Quidway -ospf-1-area-0.0.0.0] network 169.1.0.0 0.0.255.255
```

配置 LSP，即触发 PE1～PE2 之间的 LSP 的建立。

```
[PE1] mpls
[PE1-mpls] lsp-trigger all
```

4.8 MPLS L2VPN 故障诊断与排除

1. 故障现象一

故障描述：

在 VLAN 接口上配置二层 VPN 指令不成功。

故障排除：

- 检查接口上是否已经使能了 MPLS/BGP VPN、WebSwitch、组播或者 VLL 业务。若是，则不能再进行二层 VPN 配置。
- 查看 VLAN 接口是否是 Super-Vlan 或者是 Sub-Vlan。只有在正常的 VLAN 接口才能进行二层 VPN 配置。

2. 故障现象二

故障描述：

L2VPN 配置后，Ping 对端失败，查看 VC 状态，发现 VC 状态为 Down，Remote 值为无效值。

故障排除：

VC 状态为 Down 是因为两端封装能力不一致，查看本端和对端 PE 设备配置的封装类型和 MTU 是否一致，如果配置的封装能力不一样，连接将失败。

Remote 值为无效值，请检查两端是否已配置了 Remote 参数，并正确设置了对端的地址。

第5章 MPLS 流量工程

5.1 MPLS 流量工程简介

MPLS TE（Traffic Engineering）是一种将流量工程技术与 MPLS 相结合的技术。通过 MPLS TE，可以建立指定路径的 LSP 隧道，进行资源预留，并且可以进行重新优化；在资源紧张的情况下，可以根据优先级和抢占参数的情况，抢占低优先级的 LSP 隧道的带宽资源等等。同时，还可以通过备份路径和快速重路由技术，在链路或节点失败的情况下提供保护。

使用 MPLS 流量工程，网络管理员只需要建立一些 LSP 和旁路拥塞节点，就可以消除网络拥塞。随着 LSP 数量的增长，还需要使用一些特殊的离线工具进行业务量分析。

MPLS 流量工程项目前主要实现了 CR-LSP（Constraint-based routing LSP）隧道的创建和删除，能够充分利用网络现有资源，避免由于路由协议造成的链路过使用和欠使用状况。

流量工程中使用了扩展的 IGP 来实现链接状态信息的发布（OSPF 是通过扩展不透明 LSA 来实现的），加到 IGP 中的流量工程信息主要包括最大链路带宽、最小预留带宽、当前预留带宽、当前带宽使用、链接着色（即接口属性标记）等等，这些链路状态信息扩散到每一个 LSR，并存储到流量工程数据库（Traffic Engineering Data）中。起始 LSR 通过对 TED 中的信息使用约束最短路径优先（CSPF）算法来决定每条 LSP 的物理路径，在计算最短路径的同时考虑了确定的约束条件（包括 TED 中与网络资源有关的特性及用户的设置等），最终 CSPF 计算的结果传递给信令－－扩展的资源预留协议（RSVP），由 RSVP 负责建立 LSP，并能够使在 LSP 中的 LSR 预留资源。

为了实现流量工程，策略是非常重要的一个方面。MPLS 流量工程需要下列策略的支持：

(1) 备份策略

当前 CR-LSP 隧道的备份策略是建立与基本 CR-LSP 路径完全不同的备份路径。这些 CR-LSP 只与基本 CR-LSP 具有相同的入口和出口节点，并且备份支持 Ordinary 方式。

(2) 容错策略

- 基本 CR-LSP，没有快速重路由，没有备份 CR-LSP；
- 基本 CR-LSP，备份使能；
- 基本 CR-LSP，快速重路由使能；

- 基本 CR-LSP，备份和快速重路由都使能。

(3) 抢占策略

(4) 重试策略

5.2 MPLS 流程工程的配置

MPLS TE 的配置包括如下步骤：

- 使能 MPLS TE 特性，配置全局参数
- 使能 RSVP-TE
- 配置 MPLS TE IGP 协议（OSPF）
- 配置接口 MPLS TE 参数
- 配置 MPLS TE 隧道的显示路径及参数

每台设备都经过上述配置后，则该链路就具备了 MPLS TE 特性。

(1) 配置 RSVP-TE

- 使能 RSVP-TE 特性
- 启动 Hello 机制，用来检测邻居的存在
- 设定允许 Hello 消息丢失的最大次数（可选）
- 设定 Hello 消息刷新时间间隔（可选）
- 设置 PSB 和 RSB 的超时倍数（可选）
- 启动预留确认机制（可选）
- 设置处于阻塞状态的超时倍数（可选）
- 设置节点的 Path、Resv 消息的刷新周期（可选）

(2) 配置接口参数

- 配置 MPLS TE 接口最大使用带宽
- 配置 MPLS TE 接口最大可预留带宽
- 配置泛洪阈值
- 配置接口属性标记
- 配置 TE 链路的 IGP 花费

(3) 配置隧道的显示路径及隧道参数

- 配置 MPLS TE 显示路径
- 配置隧道显示路径
- 指定建立隧道使用的资源预留类型
- 指定建立隧道时支持路由记录和标记记录
- 指定建立隧道时支持环路检测

- 指定建立隧道使用的信令协议类型为 RSVP
- 设置隧道重试建立的次数
- 指定重建隧道的间隔时间
- 提交隧道当前配置

5.2.1 使能 MPLS 流量工程并配置 MPLS TE 全局参数

1. 使能 MPLS 流量工程

首先要配置 LSR ID，使能 MPLS，然后，才能执行 **mpls te** 命令，使能流量工程特性。

需在 MPLS 视图下执行该命令全局使能 MPLS 流量工程特性，然后，才能在接口下使能流量工程特性。

请在 MPLS 视图或接口视图下进行下列配置。

表5-1 使能 MPLS 流量工程特性

操作	命令
使能 MPLS 流量工程	**mpls te**
去使能 MPLS 流量工程	**undo mpls te**

接口视图下关闭流量工程特性，将删除接口上所有 CR-LSP，关闭流量工程特性。

系统视图下关闭流量工程特性，将删除所有 CR-LSP 隧道，接口流量工程特性同时被关闭。

2. 配置泛洪周期

请在 MPLS 视图下进行下列配置。

表5-2 配置泛洪周期

操作	命令
配置泛洪周期	**mpls te timer periodic** *seconds*
恢复泛洪周期为缺省值	**undo mpls te timer periodic**

缺省为 180 秒。所有接口均按泛洪周期规定的时间发布链路状态更新信息。

5.2.2 配置 RSVP-TE

1. 使能 RSVP-TE 特性

如使用 RSVP 信令控制协议，需要先在系统视图下全局使能 RSVP-TE 特性，然后相应接口视图下使能 RSVP-TE 特性。

请在系统视图、接口视图行下列配置。

表5-3 使能 RSVP-TE 特性

操作	命令
使能 RSVP-TE 特性	**mpls rsvp-te**
去使能 RSVP-TE 特性	**undo mpls rsvp-te**

当在系统视图下关闭 RSVP-TE 特性时，会同时关闭每个接口上的 RSVP-TE 特性。

2. 启动 Hello 机制，用来检测邻居的存在

请在系统视图、接口视图下进行下列配置。

表5-4 启动 Hello 机制用来检测邻居的存在

操作	命令
启动 Hello 机制用来检测邻居的存在	**mpls rsvp-te hello**
关闭 Hello 机制	**undo mpls rsvp-te hello**

3. 设定允许 Hello 消息丢失的最大次数

设定连续未接收到 Hello 消息的次数，当超过此次数时判定链路故障。

请在系统视图下进行下列配置。

表5-5 设定允许 Hello 消息丢失的最大次数

操作	命令
设定允许 Hello 消息丢失的最大次数	**mpls rsvp-te hello-lost** [*times*]
恢复连续未收到 HELLO 应答消息的缺省次数	**undo mpls rsvp-te hello-lost** [*times*]

缺省值为 3。

4. 设定 Hello 消息刷新时间间隔

请在系统视图下进行下列配置。

表5-6 设定 Hello 消息刷新时间间隔

操作	命令
设定 Hello 消息刷新时间间隔	**mpls rsvp-te timer hello** *seconds*
恢复发送 Hello 刷新消息的间隔周期的缺省值	**undo mpls rsvp-te timer hello**

缺省值为 3 秒。

5. 设置 PSB 和 RSB 的超时倍数

请在系统视图下进行下列配置。

表5-7 设置 PSB 和 RSB 的超时倍数

操作	命令
设置 PSB 和 RSB 的超时倍数	**mpls rsvp-te keep-multiplier** *times*
恢复缺省值	**undo mpls rsvp-te keep-multiplier**

缺省值为 3。

6. 启动预留确认机制

请在系统视图下进行下列配置。

表5-8 启动预留确认机制

操作	命令
启动预留确认机制	**mpls rsvp-te resvconfirm**
关闭预留确认机制	**undo mpls rsvp-te resvconfirm**

缺省为关闭预留确认机制。

7. 设置处于阻塞状态的超时倍数

请在系统视图下进行下列配置。

表5-9 设置处于阻塞状态的超时倍数

操作	命令
设置处于阻塞状态的超时倍数	**mpls rsvp-te blockade-multiplier** *times*
恢复缺省值	**undo mpls rsvp-te blockade-multiplier**

缺省值为 3。

8. 设置节点的 Path、Resv 消息的刷新周期

请在系统视图下进行下列配置。

表5-10 设置节点的 Path、Resv 消息的刷新周期

操作	命令
设置节点的 Path、Resv 消息的刷新周期	**mpls rsvp-te timer refresh** *seconds*
恢复缺省值	**undo mpls rsvp-te timer refresh**

缺省的 Path/Resv 刷新周期为 30 秒。

5.2.3 配置 OSPF-TE

OSPF TE 是基于 Opaque LSA 的，在使能 OSPF TE 之前，必须使能 Opaque 能力。若需要使用 CSPF 算法计算路由，则应进行下面配置。

1. 使能进程的 Opaque 能力

请在 OSPF 视图下进行下列配置。

表5-11 使能进程的 Opaque 能力

操作	命令
使能 ospf 进程的 Opaque 能力	**opaque-capability enable**
去使能 ospf 进程的 Opaque 能力	**undo opaque-capability**

 注意：

若某个区域已经使能了 OSPF TE，则“undo opaque-capability”命令会执行失败。

2. 使能 OSPF 区域的 TE 能力

请在 OSPF 区域视图下进行下列配置。

表5-12 使能 OSPF 区域的 TE 能力

操作	命令
使能 OSPF 区域的 TE 能力	**mpls-te enable**
去使能 OSPF 区域的 TE 能力	**undo mpls-te**

 注意：

若进程还没有使能 Opaque，则“mpls-te enable”命令会执行失败。

5.2.4 配置接口参数

1. 配置 MPLS TE 接口最大使用带宽

请在接口视图下进行下列配置。

表5-13 配置 MPLS TE 接口最大使用带宽

操作	命令
配置 MPLS TE 接口最大使用带宽	**mpls te max-link-bandwidth** *bandwidth*
恢复缺省值	**undo mpls te max-link-bandwidth**

缺省为物理接口实际带宽。

2. 配置 MPLS TE 接口最大可预留带宽

请在接口视图下进行下列配置。

表5-14 配置 MPLS TE 接口最大可预留带宽

操作	命令
配置 MPLS TE 接口最大可预留带宽	**mpls te max-reservable-bandwidth** *bandwidth*
恢复缺省值	**undo mpls te max-reservable-bandwidth**

缺省为物理接口实际带宽的 0.75 倍。一个接口上所有隧道的带宽总和应小于此接口的最大预留带宽。

3. 配置泛洪阈值

请在接口视图下进行下列配置。

表5-15 配置泛洪阈值

操作	命令
配置泛洪阈值	**mpls te flooding-threshoulds** { **down** | **up** } *percent* [*percent*] ...
恢复缺省值	**undo mpls te flooding-threshoulds** [**down** | **up**]

对于 **up** 缺省阈值为 15、30、45、60、75、80、85、90、95、96、97、98、99、100，对于 **down** 缺省阈值为 100、99、98、97、96、95、90、85、80、75、60、45、30、15。

4. 配置链路管理组属性

请在接口视图下进行下列配置。

表5-16 配置链路管理组属性

操作	命令
配置链路管理组属性	**mpls te link-administrative-group** *attribute*
取消链路管理组属性	**undo mpls te link-administrative-group**

缺省为没有配置接口链路管理组属性。

5. 配置 TE 链路的 IGP 花费

请在接口视图下进行下列配置。

表5-17 配置 TE 链路的 IGP 花费

操作	命令
配置 TE 链路的 IGP 花费	**mpls te metric** *metric*
恢复 TE 链路的 IGP 花费为缺省值	**undo mpls te metric**

缺省为 1。

5.2.5 配置隧道应用的显示路径及隧道参数

1. 配置 MPLS TE 显示路径

(1) 进入显式路径视图

进入显式路径视图，可以创建或修改指定的路径。

请在系统视图下进行下列配置。

表5-18 进入显式路径模式

操作	命令
进入显式路径模式	**mpls te explicit-path** *path-name* [**enable** | **disable**]
删除该路径	**undo mpls te explicit-path** *path-name*

(2) 为显示路径插入/删除节点和修改节点

请在显式路径视图下进行下列配置。

表5-19 为显示路径插入和删除节点

操作	命令
插入一个节点	**add hop** *ip-address1* [**strict** | **loose**] [**after** *ip-address2*]
删除一个节点	**delete hop** *ip-address*
根据index插入或者修改显示路径的结点地址。	**modify hop** *ip-address1 ip-address2* [**strict** | **loose**]
指定显式路径的下一个 IP 地址	**next hop** *ip-address* [**strict** | **loose**]

缺省情况下，没有指定显示路径。配置好的显示路径应与指定隧道关联，才会生效。

2. 封装隧道为 mpls te 模式

只有将隧道封装为 MPLS TE 之后，才能进行下面的配置。

请在隧道视图下进行下列配置。

表5-20 封装隧道为 mpls te 模式

操作	命令
设置 Tunnel 接口为 CR-LSP 隧道模式	**tunnel-protocol mpls te**

隧道缺省为 GRE 模式。

3. 配置隧道应用显示路径

将预先配置好的显示路径与指定隧道相关联。

请在隧道视图下进行下列配置。

表5-21 配置隧道应用显示路径

操作	命令
配置 MPLS TE 显示路径	**mpls te path explicit-path** *pathname*
取消该路径的配置	**undo mpls te path explicit-path**

经过此项配置后，将基于以上约束计算出 LSP 的物理路径。

4. 指定建立隧道使用的信令协议类型为 RSVP

请在隧道视图下进行下列配置。

表5-22 指定建立隧道使用的信令协议类型为 RSVP

操作	命令
指定建立隧道使用的信令协议类型为 RSVP	**mpls te signal-protocol rsvp-te**

5. 配置隧道带宽

请在隧道视图下进行下列配置。

表5-23 配置隧道带宽

操作	命令
配置 TE 隧道带宽	**mpls te bandwith** *bandwith-value*
恢复缺省的隧道带宽	**undo mpls te bandwith**

TE 隧道缺省带宽为 0。

6. 指定建立隧道使用的资源预留类型

请在隧道视图下进行下列配置。

表5-24 指定建立隧道使用的资源预留风格

操作	命令
指定建立隧道使用的资源预留风格	**mpls te resv-style [ff | se]**
恢复缺省的资源预留样式	**undo mpls te resv-style**

系统缺省为 **se** 类型。

7. 指定建立隧道时支持路由记录和标记记录

请在隧道视图下进行下列配置。

表5-25 指定建立隧道时支持路由记录和标记记录

操作	命令
指定建立隧道时支持路由记录和标记记录	**mpls te record-label**
取消该配置，恢复缺省情况	**undo mpls te record- label**

缺省情况下不支持路由记录和标记记录。

8. 指定建立隧道时支持环路检测

请在隧道视图下进行下列配置。

表5-26 指定建立隧道时支持环路检测

操作	命令
指定建立隧道时支持环路检测	**mpls te loop-detection**
取消环路检测	**undo mpls te loop-detection**

缺省情况下不支持环路检测

9. 提交隧道当前配置

请在隧道视图下进行下列配置。

表5-27 提交隧道当前配置

操作	命令
提交隧道当前配置	**mpls te commit**

5.3 MPLS TE 显示和调试

在完成上述配置后，在所有视图下执行 **display** 命令可以显示配置后 MPLS TE 的运行情况，通过查看显示信息验证配置的效果。

表5-28 MPLS TE 显示和调试命令

操作	命令
显示RSVP的路径信息	**display mpls te explicit-path** [*pathname*]
显示本地创建的隧道，以及它们的参数（如优先级、带宽、入接口和出接口、状态）	**display mpls te link-administration admission-control interface** [*interface-name* | { *interface-type interface-num* }]
显示当前链路信息	**display mpls te link-administration bandwidth-distribution interface** [*interface-name* | { *interface-type interface-num* }]
显示隧道信息	**display mpls te tunnel** [**destination** *address*] [**lsp-id** { *num* | *ipaddress* | *ipaddress num* }] [**lsr-role** { **all** | **ingress** | **transit** | **egress** | **remote** }] [{ **up** | **down** }] [**name string**] [[**incoming-interface** { *interface-name* | *interface-type interface-number* }] [**outgoing-inteface** { *interface-name* | *interface-type interface-number* }] | [**interface** { *interface-name* | *interface-type interface-number* }]]
用来显示最新提交的隧道配置信息	**display mpls te tunnel commit-last** *tun-num*
显示已经建立成功的 lsp 的配置信息	**display mpls te tunnel running** *tun-num*

操作	命令
显示隧道统计信息	**display mpls te tunnel statistics**
显示配置节点信息	**view hop**

在完成上述配置后，在所有视图下执行 **display** 命令可以显示配置后 RSVP 的运行情况，通过查看显示信息验证配置的效果。

在用户视图下，执行 **debugging** 命令可对 RSVP-TE 模块进行调试。

表5-29 RSVP-TE 的显示和调试命令

操作	命令
显示 RSVP 接口的相关信息	**display mpls rsvp-te** [**interface** *interface-type interface-number*]
显示 RSVP LSP 和带宽信息	**display mpls rsvp-te established** [**interface** *interface-type interface-number*]
显示 RSVP 邻居	**display mpls rsvp-te peer** [**interface** *interface-type interface-number*]
显示 RSVP 资源预留信息	**display mpls rsvp-te resv** [**interface** *interface-type interface-number*]
显示 RSVP 路径信息	**display mpls rsvp-te path** [**interface** *interface-type interface-number*] [**destination** *ip-address* | **lsp-id** *ip-address* | **source** *ip-address* | **tunnel-id** *ip-address* | **verbose**]
打开 RSVP-TE 不同模块的告警开关	**debugging mpls rsvp-te** { **all** | **error** | **path** | **ptear** | **resv** | **rtear** | **rconf** | **hello** | **perr** | **rerr** | **encdec** | **socket** | **traffic-control** | **main** | **tool** }
关闭 RSVP-TE 不同模块的告警开关	**undo debugging mpls rsvp-te** { **all** | **error** | **path** | **ptear** | **resv** | **rtear** | **rconf** | **hello** | **perr** | **rerr** | **encdec** | **socket** | **traffic-control** | **main** | **tool** }

5.4 MPLS 流量工程典型配置举例

1. 组网需求

配置一条动态的隧道，如下图所示，该路径由 RT1、RT3、RT4 和 RT6 组成，设置带宽为 20，预留风格为 FF 风格，并启动 CSPF 算法计算路由。

2. 组网图

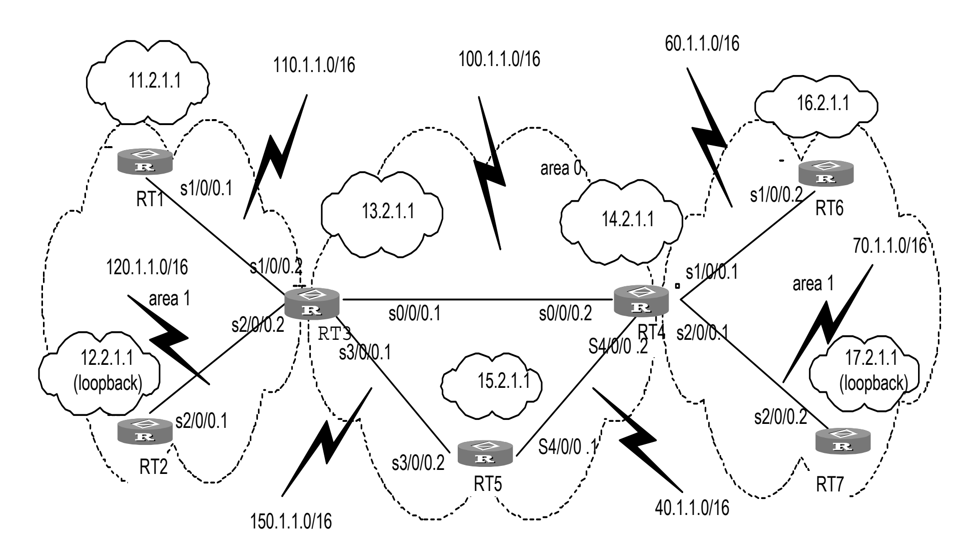


图5-1 MPLS TE 隧道典型配置举例图

3. 配置步骤

(1) 配置 RT1

```
<Quidway> system
[Quidway] sysname RT1
```

配置 LSR ID，使能 MPLS。

```
[RT1] mpls lsr-id 11.2.1.1
[RT1] mpls
```

使能 MPLS TE，使能 RSVP TE。

```
[RT1-mpls] mpls te
[RT1-mpls] mpls rsvp-te
```

配置 LOOPBACK 接口，并配置名为 huawei 的显示路径。

```
[RT1] interface LoopBack0
[RT1-LoopBack0] ip address 11.2.1.1 255.255.0.0
[RT1] mpls te explicit-path Huawei
[RT1-mpls-te-expl-path-huawei] next hop 110.1.1.2
[RT1-mpls-te-expl-path-huawei] next hop 100.1.1.2
[RT1-mpls-te-expl-path-huawei] next hop 60.1.1.2
```

配置 Serial1/0/0。

```
[RT1] interface Serial1/0/0
[RT1-Serial1/0/0] link-protocol ppp
[RT1-Serial1/0/0] ip address 110.1.1.1 255.255.0.0
[RT1-Serial1/0/0] mpls
[RT1-Serial1/0/0] mpls te
```

```
[RT1-Serial1/0/0] mpls te max-link-bandwidth 100
[RT1-Serial1/0/0] mpls te max-reservable-bandwidth 80
[RT1-Serial1/0/0] mpls rsvp-te
```

配置 tunnel1/0/0。

```
[RT1] interface Tunnel1
[RT1-tunnel1] tunnel-protocol mpls te
[RT1-tunnel1] destination 16.2.1.1
[RT1-tunnel1] mpls te path explicit-path Huawei
[RT1-tunnel1] mpls te resv-style ff
[RT1-tunnel1] mpls te bandwidth 20
[RT1-tunnel1] mpls te commit
[RT1-tunnel1] quit
```

配置 RT1 上 OSPF。

```
[RT1] ospf 1
[RT1-ospf-1] opaque-capability enable
[RT1-ospf-1] area 0.0.0.1
[RT1-ospf-1-area-0.0.0.0] network 11.2.0.0 0.0.255.255
[RT1-ospf-1-area-0.0.0.0] network 110.1.0.0 0.0.255.255
[RT1-ospf-1-area-0.0.0.0] mpls-te enable
[RT1-ospf-1-area-0.0.0.0] quit
```

(2) 配置 RT3

```
<Quidway> system
[Quidway] sysname RT3
[RT3] mpls lsr-id 13.2.1.1
[RT3] mpls
```

使能 MPLS TE，使能 RSVP TE。

```
[RT3] mpls te
[RT3] mpls rsvp-te
[RT3] interface LoopBack0
[RT3-LoopBack0] ip address 13.2.1.1 255.255.0.0
```

配置接口 Serial0/0/0。

```
[RT3] interface Serial 0/0/0
[RT3-Serial0/0/0] link-protocol ppp
[RT3-Serial0/0/0] ip address 100.1.1.1 255.255.0.0
[RT3-Serial0/0/0] mpls
[RT3-Serial0/0/0] mpls te
[RT3-Serial0/0/0] mpls te max-link-bandwidth 100
[RT3-Serial0/0/0] mpls te max-reservable-bandwidth 80
[RT3-Serial0/0/0] mpls rsvp-te
```

配置接口 Serial1/0/0。

```
[RT3] interface Serial1/0/0
[RT3-Serial1/0/0] link-protocol ppp
[RT3-Serial1/0/0] ip address 110.1.1.2 255.255.0.0
[RT3-Serial1/0/0] mpls
[RT3-Serial1/0/0] mpls te
[RT3-Serial1/0/0] mpls te max-link-bandwidth 100
[RT3-Serial1/0/0] mpls te max-reservable-bandwidth 80
[RT3-Serial1/0/0] mpls rsvp-te
```

配置接口 Serial2/0/0。

```
[RT3] interface Serial2/0/0
[RT3-Serial2/0/0] link-protocol ppp
[RT3-Serial2/0/0] ip address 120.1.1.2 255.255.0.0
[RT3-Serial2/0/0] mpls
[RT3-Serial2/0/0] mpls te
[RT3-Serial2/0/0] mpls te max-link-bandwidth 100
[RT3-Serial2/0/0] mpls te max-reservable-bandwidth 80
[RT3-Serial2/0/0] mpls rsvp-te
```

配置接口 Serial3/0/0。

```
[RT3] interface Serial3/0/0
[RT3-Serial3/0/0] link-protocol ppp
[RT3-Serial3/0/0] ip address 150.1.1.1 255.255.0.0
[RT3-Serial3/0/0] mpls
[RT3-Serial3/0/0] mpls te
[RT3-Serial3/0/0] mpls te max-link-bandwidth 100
[RT3-Serial3/0/0] mpls te max-reservable-bandwidth 80
[RT3-Serial3/0/0] mpls rsvp-te
```

配置 OSPF。

```
[RT3] ospf 1
[RT3-ospf-1] opaque-capability enable
[RT3-ospf-1] area 0.0.0.0
[RT3-ospf-1-area-0.0.0.0] network 100.1.0.0 0.0.255.255
[RT3-ospf-1-area-0.0.0.0] network 150.1.0.0 0.0.255.255
[RT3-ospf-1-area-0.0.0.0] mpls-te enable
[RT3-ospf-1-area-0.0.0.0] quit
[RT3-ospf-1] area 0.0.0.1
[RT3-ospf-1-area-0.0.0.1] network 120.1.0.0 0.0.255.255
[RT3-ospf-1-area-0.0.0.1] network 110.1.0.0 0.0.255.255
[RT3-ospf-1-area-0.0.0.1] mpls-te enable
[RT3-ospf-1-area-0.0.0.1] quit
[RT3-ospf-1] quit
```

(3) 配置 RT4

```
<Quidway> system
[Quidway] sysname RT4
```

MPLS 基本能力配置。

```
[RT4] mpls lsr-id 14.2.1.1
[RT4] mpls
[RT4] mpls te
[RT4] quit
[RT4] mpls rsvp-te
```

配置 LOOPBACK0。

```
[RT4] interface LoopBack0
[RT4-LoopBack0] ip address 14.2.1.1 255.255.0.0
```

配置接口 Serial0/0/0。

```
[RT4] interface Serial0/0/0
[RT4-Serial0/0/0] link-protocol ppp
[RT4-Serial0/0/0] ip address 100.1.1.2 255.255.0.0
[RT4-Serial0/0/0] mpls
[RT4-Serial0/0/0] mpls te
[RT4-Serial0/0/0] mpls te max-link-bandwidth 100
[RT4-Serial0/0/0] mpls te max-reservable-bandwidth 80
[RT4-Serial0/0/0] mpls rsvp-te
[RT4-Serial0/0/0] quit
```

配置接口 Serial3/0/0。

```
[RT4] interface Serial1/0/0
[RT4-Serial1/0/0] link-protocol ppp
[RT4-Serial1/0/0] ip address 60.1.1.1 255.255.0.0
[RT4-Serial1/0/0] mpls
[RT4-Serial1/0/0] mpls te
[RT4-Serial1/0/0] mpls te max-link-bandwidth 100
[RT4-Serial1/0/0] mpls te max-reservable-bandwidth 80
[RT4-Serial1/0/0] mpls rsvp-te
[RT4-Serial1/0/0] quit
```

配置接口 Serial2/0/0。

```
[RT4] interface Serial2/0/0
[RT4-Serial2/0/0] link-protocol ppp
[RT4-Serial2/0/0] ip address 70.1.1.1 255.255.0.0
[RT4-Serial2/0/0] mpls
[RT4-Serial2/0/0] mpls te
[RT4-Serial2/0/0] mpls te max-link-bandwidth 100
[RT4-Serial2/0/0] mpls te max-reservable-bandwidth 80
[RT4-Serial2/0/0] mpls rsvp-te
```

```
[RT4-Serial2/0/0] quit
```

配置接口 Serial4/0/0。

```
[RT4] interface Serial4/0/0
[RT4-Serial4/0/0] link-protocol ppp
[RT4-Serial4/0/0] ip address 40.1.1.2 255.255.0.0
[RT4-Serial4/0/0] mpls
[RT4-Serial4/0/0] mpls te
[RT4-Serial4/0/0] mpls te max-link-bandwidth 100
[RT4-Serial4/0/0] mpls te max-reservable-bandwidth 80
[RT4-Serial4/0/0] mpls rsvp-te
[RT4-Serial4/0/0] quit
```

配置 OSPF。

```
[RT4] ospf 1
[RT4-ospf-1] area 0.0.0.0
[RT4-ospf-1-area 0.0.0.0] network 40.1.0.0 0.0.255.255
[RT4-ospf-1-area 0.0.0.0] network 100.1.0.0 0.0.255.255
[RT4-ospf-1-area 0.0.0.0] mplste enable
[RT4-ospf-1-area 0.0.0.0] quit
[RT4-ospf-1] area 0.0.0.1
[RT4-ospf-1-area 0.0.0.1] network 60.1.0.0 0.0.255.255
[RT4-ospf-1-area 0.0.0.1] network 70.1.0.0 0.0.255.255
[RT4-ospf-1-area 0.0.0.1] mplste enable
[RT4-ospf-1-area 0.0.0.1] quit
[RT4-ospf-1] quit
```

(4) 配置 RT6 上

```
<Quidway> system
[Quidway] sysname RT6
```

MPLS 基本能力配置。

```
[RT6] mpls lsr-id 16.2.1.1
[RT6] mpls
[RT6] mpls te
[RT6] quit
[RT6] mpls rsvp-te
[RT6] interface LoopBack0
[RT6-LoopBack0] ip address 16.2.1.1 255.255.0.0
[RT6-LoopBack0] quit
```

配置接口 Serial1/0/0。

```
[RT6] interface Serial1/0/0
[RT6-Serial1/0/0] link-protocol ppp
[RT6-Serial1/0/0] ip address 60.1.1.2 255.255.0.0
```

```
[RT6-Serial1/0/0] mpls
[RT6-Serial1/0/0] mpls te
[RT6-Serial1/0/0] mpls te max-link-bandwidth 100
[RT6-Serial1/0/0] mpls te max-reservable-bandwidth 80
[RT6-Serial1/0/0] mpls rsvp-te
[RT6-Serial1/0/0] quit
```

配置 OSPF。

```
[RT6] ospf 1
[RT6-ospf-1] area 0.0.0.1
[RT6-ospf-1-area 0.0.0.1] network 16.2.0.0 0.0.255.255
[RT6-ospf-1-area 0.0.0.1] network 60.1.0.0 0.0.255.255
[RT6-ospf-1-area 0.0.0.1] mplste enable
[RT6-ospf-1-area 0.0.0.1] quit
```

5.5 MPLS 流量工程故障诊断与排错

故障之一：无法产生描述 MPLS TE 信息的 TE LSA。

故障排除：打开 OSPF TE 的调试开关，具体操作请参见 **debugging** 命令中的 **debugging ospf te**。根据路由器上的调试信息，初步定位错误，然后使用其它命令作进一步的判断。如果发现 OSPF 没有收到建立 TE LINK 的消息，则可能是还没有配置 MPLS TE 或者相关接口没有 UP。否则，可以检查 OSPF 邻居是否正常建立。